ASSESSMENT OF APPLICATIONS FOR THE HOUSE RENOVATION GRANTS SYSTEM

An IT Support Framework

FRANCISCO J. LOFORTE T. RIBEIRO

A THESIS SUBMITTED TO THE UNIVERSITY OF SALFORD FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF SURVEYING
UNIVERSITY OF SALFORD

JULY 1995

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

This research was supported by grants from "Programa Ciência" and "Programa Praxis XXI" of the "Junta Nacional de Investigação Científica e Tecnológica" (JNIC) of Portugal

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to Professor Peter Brandon for his most appreciated supervision, guidance and support.

I also would like to thank the JNIC (Junta Nacional de Investigação Científica e Tecnológica) for the financial support to the research.

I wish to thank the Department of Civil Engineering, the Construction Economics and Management Group and the Scientific Committee of the IST (Instituto Superior Técnico) in Portugal for their continuing support to my research.

I wish to extend my deep gratitude to the staff of the Department Of Environmental and Consumer Services of the Salford City Council for their support and help to this research, especially to Mr. Henry Tweedy and Mr. Chris Linghard.

I wish to thank the staff of the Department of Surveying of University of Salford for the support to my work, especially to Professor Peter Barrett, Dr. Ian Watson, Mark Picton, Andrew Cody, Catherine Stanley, Jonathan Holling and Martin Sexton.

Many thanks to PhD students of Surveying, especially to Bridget Palmer, Steve Allen and Eduardo de Sousa.

Finally, I wish to say a very personal thank you to my family for their continuous support and friendship.

ABSTRACT

This thesis describes the development of a knowledge-based framework for supporting human experts from the Department of Environmental and Consumer Services of the Salford City Council in assessing applications for the current house renovation grant system (HRGS).

The resulting system implements an architecture which integrates case-based reasoning processes with other problem solving methods. In addition, the system's architecture integrates different types of knowledge which are required by the problem solving methods. Some of the main features of the system's architecture are its modularity and its independence from the implementation shell. The system was implemented using Kappa-PC which is a shell designed for implementing knowledge-based systems.

The implementation followed the Client Centred Approach method. A number of lessons were learned from the implementation process. The implementation was carried out together with the verification and validation of the system. The verification and validation methods employed allowed the author to focus the evaluation on different features and components of the system. A number of test cases were employed during the validation. Client's experts and other independent experts were involved in the validation of the system. Each validation step was followed by refinement of the main system's components.

This research has demonstrated that various problem solving methods are required for performing the different tasks of the assessment of applications for the HRGS. The implemented system has been reasonably successful in demonstrating that a single framework which integrates various methods can be used for supporting human experts in assessing applications for the HRGS. Therefore, the system has proved to perform as accurately as human experts do for all of the tasks. The system has been described as very promising by the Client's experts.

CHAPTER 1

INTRODUCTION

1.1- CHOICE OF SUBJECT FOR RESEARCH

A key aspect for discussion prior to the commencement of this research was the subject of research within the broad area of information technology (IT) in support of the construction industry.

Brandon (Brandon P., 1993a) has pointed to *intelligence* and *integration* as being the key concepts for the development of IT in the construction industry over the next decade. Brandon (Brandon P., 1993a) referred to *intelligence* in the sense of using IT for the development of knowledge-based systems for delivering the right information, at the right time and in the right format to support human decision making in the construction industry.

A knowledge-based system (KBS) is essentially a computer program designed to represent and use expertise and knowledge of a variety of problems (Buchanan B. and Smith R., 1993). Brandon (Brandon P., 1993a) mentioned five major contributions which KBSs can provide for the user within the construction industry. These include:"

- The derivation of a quick first hypothesis or starting point for problem solving based on past experience from leading practitioners.
- Consistency and integrity in the knowledge base to avoid error, particularly when exercising 'what if' functions.
- Supporting explanations for assumptions made- although the level of explanation is still very limited.
- Intelligent prompts for when human decision making is more appropriate.
- Selectivity in choosing the information support (e.g. from a database needed by a consultant or contractor)".

A number of KBSs which have already been deployed or are in the development stage have proved how some of the concepts and models of artificial intelligence (AI) (such as: knowledge-based systems, case-based reasoning, rule-based reasoning, model-based reasoning, pattern recognition, etc.) can be usefully combined with other computer technologies to produce and deliver powerful decision-aiding systems in the construction industry. Such KBSs include: ELSIE (Brandon P. et al., 1988); Construction PLANNEX

(Zozaya-Gorostiza C. et al., 1989); EMMY (Watson I. and Brandon P., 1993); CASHFLOW (Brandon P., 1993a); CADSYN (Maher M. and Balachandran B., 1994); ARCHIE (Pearce M. et al., 1992); MEMORABILIA (Oxman R.E., 1991); CYCLOPS (Navinchandra D., 1992); and STRUPLE (Zhao F. and Maher M., 1988). In particular, CADSYN, ARCHIE, MEMORABILIA and CYCLOPS are some of the available examples which provide evidence of the growing popularity of case-based reasoning (CBR) within the construction industry research community. They are also indications of the potential viability of CBR methods for developing KBSs in specific areas within the construction industry.

The majority of CBR applications projects in the construction industry are in architectural design and tutoring (ARCHIE; MEMORABILIA), structural design (CADSYN; STRUPLE) and landscape design (CYCLOPS). Most of these systems are still at research level. As described later in chapter 3, these systems do not address all of the main issues of CBR. This has led to the conclusion that, although there is an increased interest in CBR in the construction industry, it is still a relatively unexplored area of research which might present new challenges and opportunities. This conclusion has led the author to choose to investigate the role and application of CBR methods in a specific area of the construction industry. Kolodner (Kolodner J., 1993) pointed out that CBR methods are still relatively young and they have much promise, although there have been few industrial-strength systems built to date.

1.2- AREA OF APPLICATION IN CONSTRUCTION INDUSTRY

1.2.1- APPLICATION DOMAIN

One of the potential application areas for CBR in the construction industry is the housing renovation grant system (HRGS) for the repair and improvement of housing within the private sector.

The HRGS came into effect in 1990, through the Local Government Housing Act 1989 (Act 1989) (Parts VII, VIII and IX). It was designed as a unitary grant to suit different needs, to target resources on the houses most in need of repairs and the householders who most need help. Under the HRGS, a Local Authority (LA) is under an obligation to give renovation grants if houses are deemed unfit under the new fitness standard, to bring them up to the required standard. In addition LAs have discretionary powers to award renovation grants where works go beyond the requirements of making fit for human habitation, but which are necessary to bring the dwelling up to a 'target' standard.

One of the principal features of the HRGS is that, in relation to each application for a renovation grant (other than for the provision of dwellings by conversion), LAs are under a duty to determine whether a dwelling is fit for human habitation and, if it is not, whether renovation is the most satisfactory course of action.

Standards of the condition of the property on completion of grant aid works are fundamental to the HRGS. Any grant scheme should bring about an improvement in the condition of the property. Whether a grant is mandatory or discretionary, a test of financial resources will decide how much the applicant is able to afford to pay himself and how much of the cost of the works can be covered by a grant. If the income of the relevant persons to an application is low then the applicant may receive a grant to cover the full cost of the works. The higher the applicants' income, the more he or she will be expected to pay, possibly up to the full cost of the works.

In some LAs the activities related with the implementation and management of the HRGS are immense. In a survey conducted as part of the present research covering 60 LAs (described in chapter 2) shows that the average number of applications received monthly in the majority of LAs is between 50 and 100. In about 24% of the responding LAs, the number of monthly grant applications is between 101 and 300. Some 74% of the number of monthly grant applications are from unfit properties, making them eligible for mandatory renovation grants. Around 86% of the responding LAs, use computers to assist in some activities related to the administration of the HRGS.

IT has shown a number of advantages both in administering the HRGS and in assessing grant applications. In administering the HRGS, the computer is used mainly in activities such as: i) recording, storing and retrieving information from grant applications; ii) recording and monitoring payments; iii) recording correspondence with applicants; iv) controlling the grant progress; and v) reporting. For assessing grant applications the computer is used mainly for means testing of the household's resources and in quite a few cases for producing schedules of work. Although the current use of computers to support some of the tasks of the assessment of grant applications the postal survey mentioned in chapter 2 has shown that over 40% of responding LAs are neutral or unhappy with the use of existing software.

One of the main concerns regarding the HRGS is to speed up the processing of hundreds of applications which LAs are confronted with every year and to achieve a reasonable solution plan for each application.

From the review of published literature and from the findings of a postal survey carried out throughout 60 LAs and interviews, it seemed that there was no KBS or other computer system in place or under development, which was capable of performing all of the tasks involved in the assessment of applications for the HRGS. Besides, it is likely that the current workload for assessing and administering the applications for the HRGS might increase over time, which in turn will put new demands on LAs.

Each LA has an extensive number of past applications which are valuable records of how the legal framework and technical knowledge were applied to particular applications previously processed. These records might contain relevant context specific knowledge revealed by the LAs staff. They also reflect the views of the LA staff on how to apply the legal framework to the existing local conditions. As a consequence, each LA has built its own body of knowledge that might constitute a valuable information resource to be applied creatively to improve the current work load derived from new grant applications.

Taking into account the apparent nature of the domain, and the lack of a system in place to assist LA experts in performing all of the tasks, the assessment of applications for the HRGS was chosen as an application area for investigating the utility and validity of CBR alone or combined with other methods through the development of a KBS application.

1.2.2- SUPPORT ORGANISATION

Since LAs are in charge of the HRGS and they hold most of the available body of knowledge in the related context it is natural that this research must be carried out with the collaboration of a relevant LA body. Thus, the Environmental and Consumer Services Department of Salford City Council was chosen as the supporting organisation and designated in this research as the Client. This choice was based on the following reasons:

- the Client has shown interest in the research from the beginning of the first consultations;
- the Client has accumulated an immense experience on the implementation of the HRGS and has a huge number of processed past grant applications;

- the Client has a clear housing renewal strategy for the private sector and they have developed specific procedures concerning the use of powers that the legal framework allows; and
- there are a number of experienced human experts in the Client who are friendly towards the use of information technology.

1.3- JUSTIFICATION FOR CASE-BASED REASONING (CBR)

After the subject and the application area of research had been chosen, a number of questions had to be addressed before any major commitment to the system implementation was made. Kolodner (Kolodner J.,1993) pointed to five questions which must be considered when designing and implementing any kind of system using CBR. They include the following:

- ♦ Should CBR be used ?
- ♦ Which tasks should CBR be used for, and for which tasks should some other kind of problem solving method be used?
- ♦ How should the interaction of the CBR method with other reasoning methods within the system be addressed?
- ♦ What sort of support is needed to make the chosen task work?
- ♦ Should the system be automated, or provide support for human experts?
- ♦ What level of support should it provide, if applicable for a human expert?
- ♦ How should the case library can be maintained over time ?

Kolodner added, that perhaps the most important issues to deal with in a system using CBR are those related to building the case library- *collecting*, *representing* and *indexing cases*. Following the Kolodner guidelines, this section will discuss the first question. The remaining questions will be discussed in subsequent chapters.

The first question to be answered is whether or not the CBR method is appropriate to the assessment of applications for the HRGS. In deciding whether or not CBR is an appropriate method , three approaches may be used:

1. Look at the advantages and disadvantages provided by the CBR method. If the advantages are preferred and its disadvantages are inconsequential for a task, then CBR may be a viable method to accomplish the task (Kolodner J.,1993).
2. Compare the advantages of CBR over other alternative methods (Kolodner J., 1993).

3. Use criteria to select CBR among other alternative problem solving methods (Chandrasekaran B., 1990).

The second and third approaches are similar, although Chandrasekaran's approach provides the criteria for comparison. Before, proceeding with the evaluation of the appropriateness of CBR in the application domain, one further question needs to be answered:

What are the alternative methods available for the application task?

In principle, the tasks of the assessment of applications for the HRGS can potentially be solved by a number of problem solving methods. Five possible groups of problem solving methods can be identified, such as: i) decomposition methods; ii) associative methods; iii) CBR methods; iv) abductive methods; and v) algorithms. A description of how these methods are applied to the application domain (HRGS) is discussed in chapter 4 of this thesis.

Using the approach proposed by Kolodner, some of the advantages envisaged at the beginning of the research for the CBR method are presented below:

- Knowledge acquisition for the proposed system in terms of application cases can be a quick and easy process. According to Hennessy and Hinkle (Hennessy D. and Hinkle D., 1992), case acquisition processes are easier than traditional rule-based systems since it is easier for human experts to remember and describe specific experiences than a fixed set of production rules. Besides, the acquisition of the specific knowledge stored in past grant application is a simple process.
- The number of available past applications at the client's organisation is very large.
- CBR can provide the system with the ability to automatically acquire new application cases by adding to the case library all successfully processed grant applications. By doing this, CBR allows a system to maintain and update the knowledge-base without having to restructure it manually, as can happen in rule-based systems (Hennessy D. and Hinkle D., 1992).
- CBR can provide the system with the ability to derive quick and reliable solutions for some of the tasks of the assessment of applications for the HRGS. This is an inherent advantage of the CBR method over other methods that derive solutions from scratch, such as the associative and heuristic methods (Chandrasekaran B. et al., 1992; Kolodner J. 1993; Leak D., 1991).

- CBR can help to get the application running quickly, because it can work with a small case library. For example, systems such as ARCHIE and CLAVIER started working with only 20 cases stored in their case libraries.
- CBR may be suitable for capturing and modelling the large body of knowledge contained in past applications in terms of cases. Hennessy and Hinkle (Hennessy D. and Hinkle D., 1992) referred to the unique ability of CBR systems to capture and apply corporate knowledge.

Some of the disadvantages that can be envisaged are as follows:

- Application cases in the case library need to be validated in terms of its content and structure before being used for solving new problems. This problem was pointed out by Simoudis (Simoudis E., 1992) regarding the case-based system CASCADE. Kolodner and Mark (Kolodner J. and Mark W., 1992) emphasised that without sufficient validation the system can retrieve inappropriate cases.
- According to Kolodner (Kolodner J., 1993), for CBR to be useful and reliable, cases with similar problem descriptions should have similar solutions.
- Case libraries require considerable storage space as they grow over time.
- Real-world cases can be large in size or incomplete in terms of their content.
- CBR may not be suitable for some tasks of the assessment of applications for the HRGS.

The advantages and disadvantages have been presented, but in order to make a final conclusion it was decided to also use Chandrasekaran's approach, which is based in the following criteria (Chandrasekaran B., 1990):

- ♦ **Solution effectiveness:** How effective are the various alternative methods at solving different variants of grant applications?
- ♦ **Computational Efficiency:** How computationally efficient are the various methods at solving the assessment task? Is the computation feasible? How much time does it take to reach a solution? How transparent is the reasoning process to the user?
- ♦ **Knowledge Availability:** What types of knowledge are available in the given house renovation grant domain? How well do the available types of knowledge match the types of knowledge required by the various methods?

Applying at this stage part of the Chandrasekaran's criteria (Chandrasekaran B., 1990), the following conclusions were drawn:

- Regarding the first criteria, CBR may help the system to derive accurate solutions for certain tasks of the assessment of applications for the HRGS assuming those application cases in the case library and the retrieval functions are validated. According to Goel (Goel A., 1989) associative and heuristic methods require highly compiled forms of knowledge that can be represented as production rules. Due to the youth of the HRGS and the discretion left by the legal framework (presented in chapter 2) it is difficult to formulate domain rules for some tasks. Existing application cases help to fill in the knowledge gaps in the domain.
- Regarding the second criteria, CBR may help the system to derive solutions much quicker and with less computation than other methods. In similar situations it can derive solutions by retrieving or retrieving and adapting an old solution. Associative and heuristic methods derive solutions from scratch requiring more computation and more knowledge in compiled form for exploring the problem spaces (Goel A., 1989; Kolodner J., 1993).
- Regarding the third criteria, knowledge in the form of past applications at the client's organisation is abundant in range of problems covered. Domain knowledge in terms of rules is difficult to find in the HRGS.

Taking into consideration the analysis (advantages against disadvantages and the criteria), it was concluded that there is enough evidence pointing towards the use of CBR for some tasks of the assessment of applications for the HRGS.

1.4- MOTIVATION FOR RESEARCH WORK

The primary motivation was the attraction created by the challenges and opportunities that CBR promises for the construction industry and the optimism shown by the whole community of researchers in CBR. As referred previously in this chapter most case-based systems in the construction industry are still at research level. Thus, there are several kinds of challenges with immense opportunities for designing and developing simple systems which can operate with advantage in the construction industry and in particular for the HRGS. The HRGS domain is largely documented, with a number of applications routinely being assessed and new knowledge coming in as the result of the experience. This provides many opportunities in the search for better IT based decision aiding tools which might bring about real benefits to the work being carried out by the LAs. CBR models can contribute to achieving that goal.

The second motivation was the interest of the author in housing renovation. The author comes from an old city- Lisbon- with many renovation problems. One of the main causes for the present state of dilapidation of a huge number of buildings in Lisbon is the lack of appropriate means and tools to deal with the problem. The poor quality of housing is high especially in the private sector. The study of the HRGS in England and its rich experience could be very useful for the author.

The third motivation is the increasing importance of the HRGS within the context of the housing renewal policy for private housing. According to the English Housing Condition Survey 1991 (EHCS91) (Department of the Environment, 1991), at the end of 1991 there were 19.7 million dwellings in England and almost half of this stock was more than 50 years old. A quarter of the stock was built before 1919. Under the new fitness standard some 7.6% of dwellings in England were found to be unfit for human habitation. Within the global housing renewal strategy, the HRGS seems to be an important tool at the hands of the LAs to deal with unfit properties and to improve the human housing conditions of households in the private sector: owner-occupied and rented.

Finally, the last motivation is the background of the author, with over 25 years of experience in the construction field including the renovation of buildings. Therefore, the author has developed a KBS prototype for his MSc degree in another domain.

1.5- AIMS OF RESEARCH

1.5.1- RESEARCH HYPOTHESES

On the basis of the discussion presented in previous sections, the hypotheses of this research are outlined as follows:

1. *Case based reasoning can provide the means for the successful collecting, modelling, representing and indexing of relevant information from grant applications into a case library of application cases. This information includes: i) the description of the application; ii) the solution plan for the application; and iii) the outcomes of the grant application.*

2. *Case-based reasoning can enable the proposed system to derive, by retrieving or adapting solutions contained in past application cases stored in the its case library, schedules of work based on the assessment of the condition of the house with respect to the standard of fitness for human habitation. Therefore, according to the type of grant, these schedules of works should be for: i) making fit an unfit property; ii) adapting a house to meet the adaptation needs of a disabled occupant; iii) repairing or improving a house to meet the needs of elderly occupants.*

3. *Case-based reasoning can enable the proposed system to derive with accuracy from past applications stored in its case library, a consistent and complete solution plan for a new grant application based on the economic analysis of the renovation action, means test, and eligibility of the application.*

4. *Case-based reasoning can enable the proposed system to learn from its problem solving experiences by automatically indexing and adding to its case library any new grant application case successfully solved. Such features can also allow the system to maintain and up-date the case library without having to restructure it.*

These hypotheses will be tested and evaluated through the implementation, verification and validation of the proposed system. The conclusions are presented in chapter 9.

1.5.2- OBJECTIVES

This research was carried out by developing a practical application through which the hypotheses will be assessed. The specific objectives of such development are presented below:

1. To understand the nature of expertise and skills employed by human experts when assessing the grant applications and to find out how much of this expertise can be modelled in a KBS using different problem solving approaches. Also, to analyse the nature of knowledge contained in the past applications.
2. To investigate the role and benefits of using CBR in the assessment of house renovation grant applications.
3. To investigate the feasibility and utility of task-based approaches for modelling the system at the knowledge-level and the problem space level.

4. To develop an architecture which supports human experts in assessing grant applications for the HRGS.
5. To investigate whether or not cases are an adequate means for modelling, representing, indexing and presenting relevant information stored in the grant applications.
6. To implement a working KBS potentially useful for supporting human experts in the assessment of grant applications for the HRGS.

1.6- SCOPE

To ensure that the research work would be completed before the ending of the funding period (set by JNIC- "Junta Nacional de Investigacao Cientifica e Tecnologica"- of Portugal), the scope was clearly defined at the beginning of the work. The scope of research in the chosen subject and application area was defined with respect to classes of grant application problems and domains.

1.6.1- LIMITATION ON CLASS OF APPLICATION PROBLEMS

This research and the proposed system are limited in their applicability to a class of application problems that can be characterised as follows:

- Well documented grant applications with respect to their description.
- Applications from both types of applicants, singles and couples.
- Applications from employed earners, pensioners and persons on income support.
- Applications regarding the repair or improvement of houses.
- Applications from owner-occupier's households.

1.6.2- LIMITATION ON GRANT APPLICATION DOMAINS

This research and the proposed system are limited in their applicability to a class of application domains within the HRGS that can be characterised as follows:

- Renovation grant: the research and the system are limited to mandatory renovation grants.
- Disabled Facilities Grant: the research and the system are limited to mandatory disabled facilities grants.

- Minor Works Assistance: the research and the system are limited to minor works assistance.

1.7- RESEARCH METHODOLOGY AND DELIVERABLES

1.7.1- THE METHODOLOGY ADOPTED

The methodology that was used to conduct this research generally follows the *Task Structure Analysis* (unified framework) from Chandrasekaran, Johnson and Smith (Chandrasekaran B., et al., 1992) and the *Client Centred Approach* (CCA) from (Watson I., et al. 1992a, 1992b). Following both methods the proposed system will be described using two levels:

- **The knowledge-level:** to provide a specification of the system in terms of what the system should do and how the system accomplishes its goals. The product of this level is a task structure for the assessment of grant applications in terms of goals, sub-tasks and methods to accomplish the sub-tasks. Newell (Newell A., 1982) introduced the notion of knowledge-level and proposed a Knowledge Level (KL) framework to describe intelligent systems without being linked with the details of its implementation. The Newell's KL, was a useful contribution to the definition of the knowledge-modelling level, which may have inspired the shape of later contributions on the knowledge modelling.
- **The symbol-level:** where the system is implemented using any computer language. The products of this level are; i) the skeleton system; ii) the demo system; and iii) the working system.

The above two description levels, their products and development activities relate to them are illustrated in figure 1.1 shown below. According to this methodology, each level of description is a self contained model of the system. In this methodology the $level_n$ implements the structure of the $level_{n-1}$. The first level allows the author to describe the system without being concerned with how to implement it. This is illustrated by figure 1.2 below.

Research Methodology

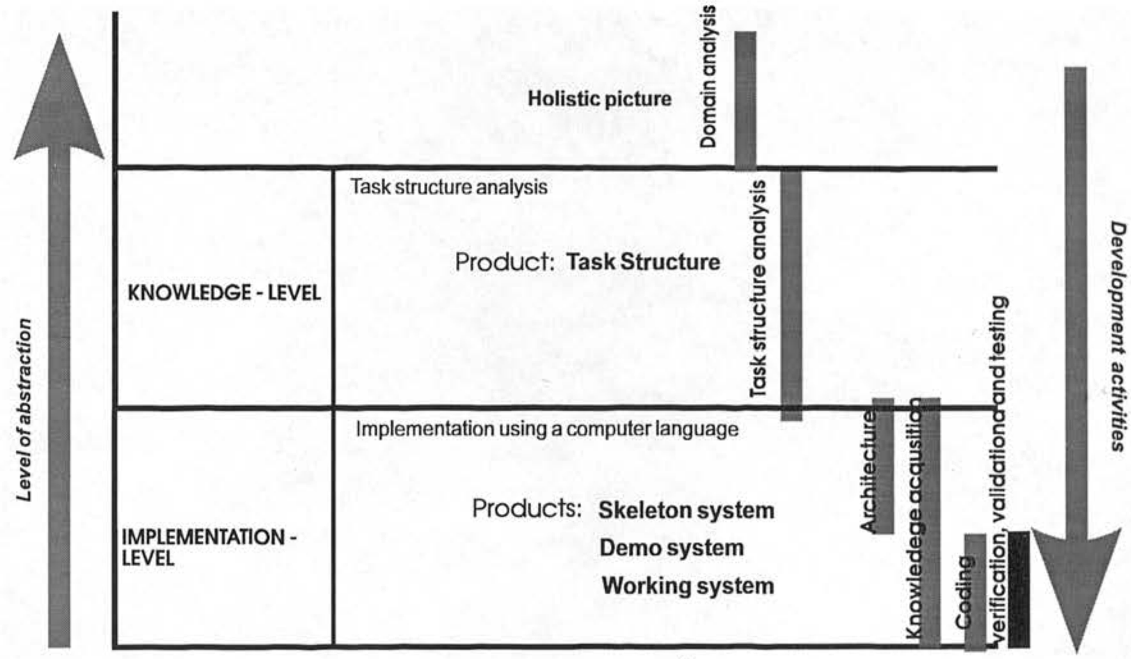


Figure 1.1: View of the research methodology

At each level of the research methodology, the following methods were followed:

- At knowledge level the *Task Structure Analysis* (unified framework) from Chandrasekaran, Johnson and Smith (Chandrasekaran B. et al., 1992) was followed. The task structure for the proposed system is outlined and discussed in chapter 4.

- To implement the proposed system at symbol level the *Client Centred Approach* (CCA) was followed (Watson I. et al., 1992a, 1992b). The Client Centred Approach is described in chapter 7.

Some major advantages are expected from the proposed research methodology. They are that:

- It describes the system by abstracting from the details of implementation in a computer language or shell.
- It allows the author to focus on the knowledge content of the system without being worried about how it will be represented and implemented.
- It facilitates the specification of the system and its design based on the expected behaviour.

- It will facilitate the verification and validation of the system because the description at knowledge-level provides a specification of the system.

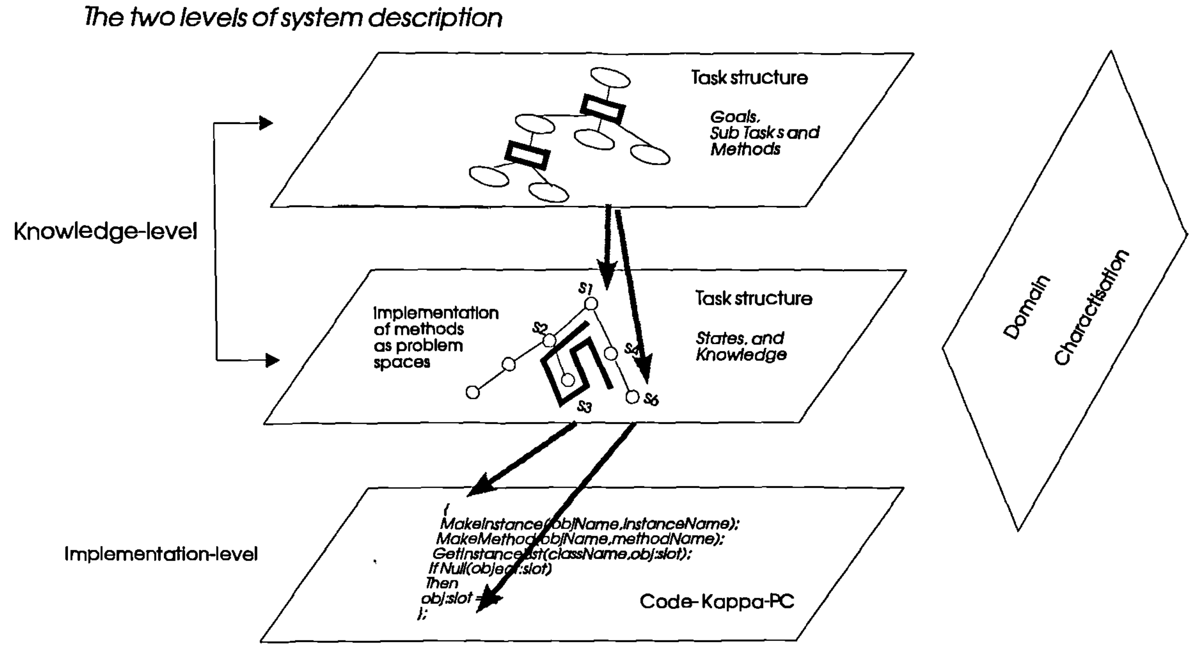


Figure 1.2: Illustrative types of descriptions used in the three levels

1.7.2- DELIVERABLE PRODUCTS

During the study and at its completion the following deliverables will be produced as a result of the methodology used to conduct this research:

1. **Task Structure:** A task structure for the assessment of house renovation applications.
2. **The System's Architecture:** The architecture for the system.
3. **Xtimela-CBR System:** A working version of the system.

1.8- OUTLINE OF THE THESIS

The thesis is divided into three main parts. The first part comprises chapters 1 to 5, which consists of: i) a characterisation of the domain; ii) a review of the systems using CBR; and iii) modelling and acquisition of the system knowledge. The second part comprises chapters 6 to 8, and focuses on the implementation and evaluation of the proposed system. Finally, the third part consists of chapter 9, containing the major conclusions and proposals for future work. Thus, the thesis is organised as follows:

- Chapter 2 presents a characterisation of the context of the HRGS and discusses the assessment problem of grant applications in terms of the purposes, legal framework, current practice and problems.
- Chapter 3 provides a general discussion on implementation issues. It reviews some of the main applications using CBR developed in several general areas and specifically in the construction industry so far. It establishes also a number of guidelines for the development of the proposed system.
- Chapter 4 outlines and discusses the task structure for the assessment of grant applications in the context of the characterisation presented in chapter 2. The task structure specifies what the proposed system should do and how it should accomplish the assessment of applications for the HRGS.
- Chapter 5 outlines and discusses the knowledge required to perform the sub-tasks in the problem spaces as discussed in chapter 4. It also provides a general description of the knowledge acquisition process followed in the context of this research.
- Chapter 6 introduces an architecture for the proposed system and covers some aspects of the application components. This architecture was used to implement the system.
- Chapter 7 concentrates on describing the implementation of the system. It also describes the main features of the implemented system.
- Chapter 8 discusses the problems of verifying and validating the proposed system, as well as describing the approach followed in the current research.
- Finally, chapter 9 summarises the conclusions of the present research, lessons for the future and suggestions for future work. It concludes this thesis.

Throughout this thesis, a number of expressions are widely employed from the field of HRGS and artificial intelligence, such as means test, mandatory grant, task structure, methods, problem space, search space, states, rules, objects, cases, adaptation. The meanings of such expressions have been defined in several publications. They can also be found in the glossary of expressions and terms presented in Appendix 1. The meaning of the abbreviations used in this thesis can also be found in the list of abbreviations presented in Appendix 2.

CHAPTER 2

CHARACTERISATION OF THE HOUSE RENOVATION GRANT SYSTEM

2.1- INTRODUCTION

In chapter 1 CBR has been pointed out as one of the methods that has the potential for improving the current situation regarding the assessment applications for the house renovation grant system (HRGS). In addition, it was argued that assessment of grant applications is a complex problem solving activity, and the HRGS domain is wide.

This chapter presents a general characterisation of the HRGS domain which provides a basis for modelling the system proposed in this research.

In building a KBS, the *overall task* which the system is responsible for and the *domain* in which it is carrying out the task, together called the *task domain*, needs to be characterised (Kolodner J., 1993). Motoda (Motoda H., 1994) pointed out that a constructive approach for modelling a KBS in terms of the knowledge it contains must be guided by a characterisation of the domain where the task is performed. Thus, before the proposed system will be modelled in terms of the task and sub-tasks it is responsible for and the knowledge it might contain, a characterisation of the domain needed to be undertaken in the context of this research.

This chapter initially presents a general description of the HRGS domain as well the legal context where the overall task is performed. The second part of this chapter discusses some issues related with the current situation of the assessment of applications for HRGS. The description presented in the first part is the result of a detailed review of the governmental publications concerned with the HRGS. The conclusions outlined in the second part are based on a postal survey carried out at the beginning of this research.

A postal survey was carried out covering 60 LAs with the aim of characterising the current situation in the HRGS domain. A total of 31 questionnaires were completed by the LAs targeted, which gives a response rate of 52%. The data obtained from the questionnaires was later processed using the SPSS/PC software package. The questionnaire form used to collect the data is presented in Appendix 3.

2.2- THE HOUSE RENOVATION GRANT SYSTEM (HRGS)

2.2.1- THE GOAL OF A GRANT APPLICATION

The condition of housing stock and the quality of the accommodation it provides is determined in part by sustained expenditure on its repair and or improvement. According to the EHCS91 (Department of the Environment, 1991), at the end of 1991 some 1.9 million households were living in the worst category of housing conditions. Over 36% of these households were unhappy with the state of repair of their homes. The EHCS91 has identified three distinct priority areas of the households living in the worst condition category:

- their home;
- their immediate environment; and
- their wider neighbourhood.

Under the Local Government and Housing Act 1989 (Act 1989) a property on which any type of grant is given must, on completion of the relevant works, achieve the standard of fitness for human habitation. Following this definition, two goals must be met by a renovation grant. These goals are:

- Any renovation grant scheme should bring about an improvement in the condition of the property to a specified standard, and consequently to improve the human condition of its households. Standards refer to the condition of a property on completion of grant-aided work.
- Available resources are targeted at those which have the greatest need regarding to their financial situation.

These goals are illustrated by figure 2.1.

Taking into consideration the above goals then the overall goal of the assessment of a grant application can be formulated as follows:

A solution plan containing sufficient information about the renovation or adaptation action so that it can be successfully implemented in order to bring about an improvement in a property up to a specific standard of fitness for human habitation and target those people in greatest need.

The procedure for any house renovation grant commences formally with the making of an application after an initial enquiry. Once a valid formal application has been made LAs are obliged to consider it. An authority could be open to challenge in the courts if they were to refuse, to entertain a valid application, or to comply with any reasonable request by a potential applicant. A LA must approve or refuse the application within six months of the date it is made.

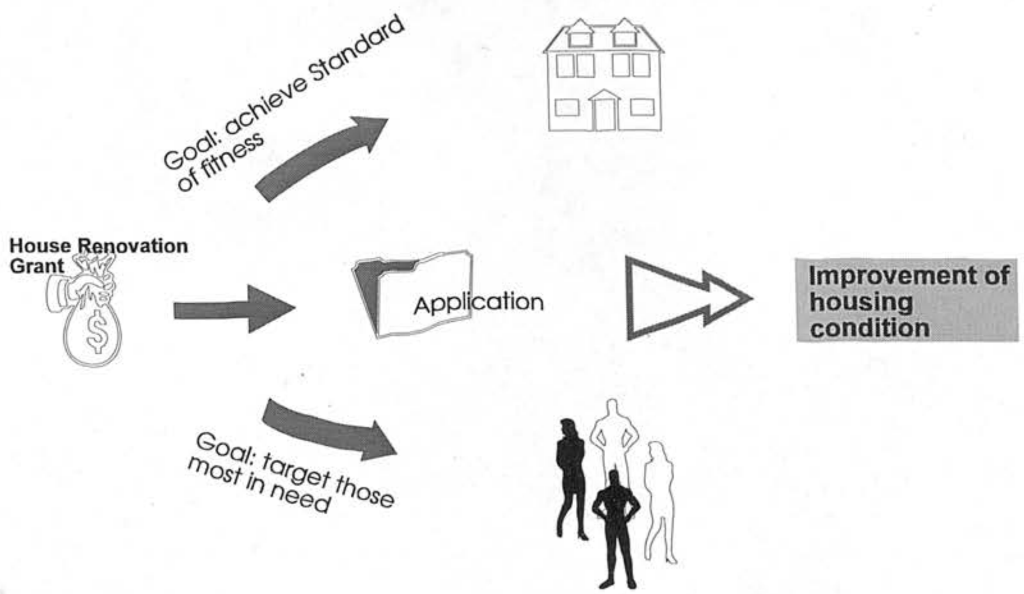


Figure 2.1: The goals of a renovation grant

2.2.2- THE HOUSE RENOVATION GRANT SYSTEM (HRGS)

Part VII and Part VIII of the Act 1989 (Act 1989) introduced a set of provisions which are intended to form a comprehensive package of powers to tackle disrepair mainly in the private sector. This package of powers underlies a housing renewal unified strategy and provides means to support its main objective of securing a reduction in the number of unfit houses, whether by repair and improvement or demolition. Such a renewal strategy includes the basic interventions of: i) Renewal Areas; ii) Group Repair Schemes; iii) Renovation Grants; and iv) Slum Clearance (Department of the Environment, 1990b).

More specifically, the Part VIII of Act 1989 introduced a new house renovation grant package, consisting essentially of five types of grants. The type of grants available in the HRGS are as follows (Department of the Environment, 1990b):

- *Renovation grants*: for the improvement and or repair of houses (including maisonettes and flats), and for the conversion of houses and other buildings into flats for letting.
- *Disabled facilities grant*: for adapting, or providing facilities to, the home of a disabled person in order to make it more suitable for him or her to live in. The disabled facilities grant is available for adaptations to, or for providing facilities in, the common parts of buildings containing one or more flats.
- *Common parts grant*: for the improvement and or repair of the common parts of buildings containing one or more flats.
- *House in multiple occupation grant*: for the improvement and or repair of Houses in multiple occupation (HMO) and for the conversion of buildings into HMO.
- *Minor works assistance*: in addition to the grants outlined above, minor works assistance is available for carrying out small-scale works.

Table 2.1 and figure 2.2 show the distribution of applications by the type of grant sought since the Act 1989 became effective.

Table 2.1: Grants paid under Act 1989 in England an Wales by type of grant (Department of the Environment 1982-1992, 1993)

Number of grants

Year	All Grants	Renovation Grants		Disabled Fac. Grants		HMO Grants		Common Parts		Minor works assist.	
		All Grants	Percentage	All Grants	Percentage	All Grants	Percentage	All Grants	Percentage	Assistance	Percentage
	numbers	numbers		numbers		numbers		numbers		numbers	
1990	7,563	514	7%	373	5%	5	0%	4	0%	6,667	88%
1991	59,470	18,192	31%	10,970	18%	383	1%	883	1%	29,042	49%
1992	81,293	34,745	43%	16,548	20%	815	1%	1,032	1%	28,153	35%
1993	88,408	43,953	50%	17,158	19%	1,195	1%	256	0%	25,846	29%

Source : Housing and Construction Statistics, HMSO, 1990, 1991, 1992, 1993
1- HMO stands for Houses in Multiple Occupation
2-The new system of grants under Local Government and Housing Act 1989 came into operation from July 1990,apart from Minor Works Assistance (April 1990).

Table 2.1 and figure 2.2 show that the renovation grant, disabled facilities grant and the minor works assistance were the predominant types of grant between 1990 and 1993.

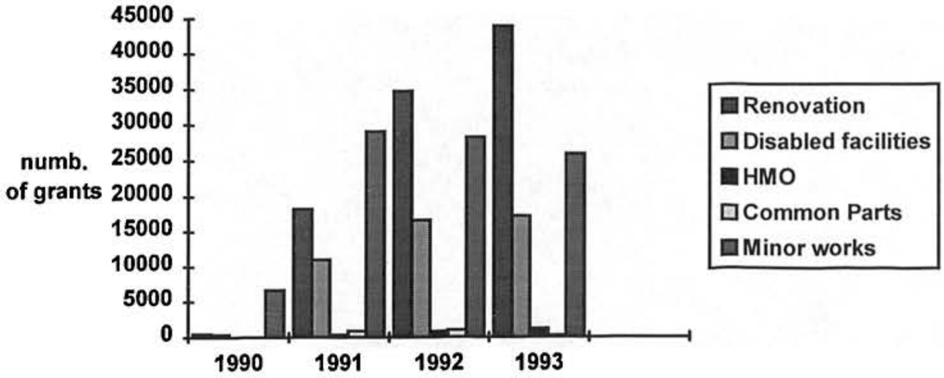


Figure 2.2: Distribution of house renovation grants by type since 1990

A detailed description of each house renovation grant which is within the scope of this research is provided in the following sections.

2.2.2.1- Renovation Grant

Although the renovation grant is designed for a wide range of purposes, its key aim is to ensure that unfit properties are made fit on completion of grant-aided works at an acceptable and economically feasible cost.

In relation to each application for renovation grants (other than for the provision of dwellings by conversion) a LA will be under a duty to determine whether a dwelling is fit for human habitation. If the property is not fit, then the LA will have to consider whether renovation is the most suitable course of action. Local authorities are encouraged to adopt a long-term view of individual properties in giving both mandatory and discretionary grants. They will also wish to bear in mind the recommendations in the Code of Guidance (Department of the Environment, 1990a) on matters such as: i) the cost of the works; ii) and proposals for the future of the area in which the premises are located.

Renovation grants are mandatory when given towards eligible works required to bring a dwelling up to the fitness standard or to comply with a statutory notice. Table 2.2 shows the number of mandatory and discretionary grants approved between 1990 and 1993.

Table 2.2 shows the predominance of mandatory renovation grants.

The main purposes for which a renovation grant is intended are (Department of the Environment, 1990b):

- *To bring an unfit property up to the standard of fitness*: If a property falls below the standard of fitness then some form of action will be required. If the LA is satisfied that renovation is the most appropriate way of dealing with the property, a renovation grant is mandatory. LAs do not, however, have to give grants immediately if they plan to include the property in a group repair scheme within 12 months of the application date or if the property is included in a declared or planned clearance area.

Table 2.2: Renovation grants approved under the Act 1989 in England and Wales (Department of the Environment 1982-1992, 1993)

Number of grants/£ thousand

Year	Mandatory		Discretionary		All Grants numbers	Percent. of Mandatory grants	Percent. of Discretionary grants
	Grants	Amount	Grants	Amount			
1990	2,627	19,743	774	2,712	3,401	77%	23%
1991	28,752	258,049	6,072	27,250	34,824	83%	17%
1992	34,565	329,843	5,857	22,570	40,422	86%	14%
1993p	35,235	476,731	5,372	31,303	40,607	87%	13%

Source : Housing and Construction Statistics, HMSO 1990, 1991,1992, 1993
1-The new system of grants under the local Government and Housing Act 1989 came into operation from July 1990
p- provisional values

- *To repair and/or improve a property beyond the standard of fitness*: A grant is discretionary for this purpose and it can be given in addition to a mandatory grant, or on its own in the case of a fit property.
- *For home insulation*: A grant is always discretionary for home insulation.
- *For heating*: A grant is always discretionary for this purpose. It is available for providing heating facilities.
- *For providing satisfactory internal arrangements*: Grants are always discretionary for this purpose.
- *For conversion*: Grants are always discretionary for conversions.

Grant applications, other than for disabled facilities, may not be entertained in respect of properties built or provided by conversions less than ten years before the date of the application. LAs do not have to be satisfied that the applicant for a grant has the requisite interest in the land until they are actually approving the grant. Formal approval, however, cannot be given until the applicant has the necessary interest in the property. LAs may not approve any grant whether mandatory or discretionary, if the required works are completed before the application has been approved, unless the works are necessary to comply with a statutory notice.

Renovation grants given by the Client are generally to the poorest households of unfit dwellings in the private sector which includes: owner-occupied, private rented, and housing association dwellings. This means that the Client is currently approving only mandatory grants.

A grant-limit of £20,000 was centrally fixed for a mandatory renovation grant.

2.2.2.2- Disabled Facilities Grant

The disabled facilities grant (DFG) is conceived as an integral part of the HRGS, with similar underlying principles to the renovation grant. The grant is designed to help make the home of a disabled person more suitable for him or her to live in, and help the person manage more independently in the home. The disabled facilities grant may be mandatory or discretionary depending on the adaptation works proposed, and provided that various qualifying conditions are met. The adaptation works could be towards dwellings or common parts of buildings containing flats, where the disabled person is the occupant of one of the flats.

Just as local authorities are under a duty to determine whether or not a dwelling is fit when an application for renovation grant is made, a similar determination has to be made in relation to the disabled person's adaptation needs. Section 114 of the Act 1989 places a duty on LA to consult the appropriate welfare authority about the necessity of the relevant adaptations and whether they meet the needs of the disabled person. As part of the overall assessment of the application, the LA must be satisfied that it is reasonable and practicable to carry out the relevant works, having regard to the age and condition of the dwelling or building. If a candidate property is unfit, and that the relevant works will not bring it up to the required standard of fitness, then the LA may not approve a DFG to carry out adaptation works, by virtue of section 107(2). In these circumstances a renovation grant may be appropriate to make the dwelling fit before it can be adapted or facilities provided.

The works attracting a mandatory DFG are primarily aimed at (Department of the Environment, 1990b):

- *facilitating access* for the disabled person into and around his or her home;

- *adapting the controls* of and/or providing additional means of control of any heating, lighting or power supplies in order to make them suitable for use by the disabled occupant; and
- the scope of works for which a DFG is mandatory extends further to include: i) works to *facilitate access* to sleeping accommodation, or providing a new bedroom suited to the needs of the disabled occupant; and ii) works to give a disabled person *access to* parts of their home that are needed to enable them to care for someone living with them who is dependent on that care.

The table 2.3 below, shows the number of mandatory and discretionary grants approved between 1990 and 1993.

Table 2.3: Disabled Facilities grants approved under the Act 1989 in England and Wales (Department of the Environment 1982-1992, 1993)

Number of grants/£ thousand

Year	Mandatory		Discretionary		All Grants numbers	Percent. of Manda-tory grants	Percent. of Discre-tionary grants
	Grants	Amount	Grant	Amount			
1990	118	518	19	63	137	86%	14%
1991	1,854	9,055	52	177	1,906	97%	3%
1992p	2,427	10,962	362	2,475	2,789	87%	13%
1993p	22,470	95,822	664	2,482	23,134	97%	3%

Source : Housing and Construction Statistics, HMSO, 1990, 1991, 1992, 1993
1-The new system of grants under the local Government and Housing Act 1989 came into operation from July 1990
p- provisional values

Table 2.3 shows the predominance of mandatory grants over all disabled facilities grants approved between 1990 and 1993.

DFGs are wider in scope than renovation grants in that they may be given for works not only to dwellings, but also to the common parts of buildings containing flats. The grant limit for a mandatory disabled facilities grant is £ 20,000.

2.2.2.3- Minor Works Assistance

Minor works assistance is intended to complement the mainstream assistance available through the HRGS. It is designed to streamline assistance with repairs and improvements to properties upon which only small scale work is required and where full scale renovation is inappropriate. It is also designed to assist elderly people who wish to remain in their homes, by helping with repairs and improvements. Minor works assistance is always discretionary, in the form of a grant or the provision of materials, and eligibility is based on receipt of income related benefits by those to whom the grant is sought. The total amount of assistance is £1,080.00. More than one application can be submitted in respect of the same dwelling, but the total amount approved may not exceed £3,240 in any three year period.

2.2.3- THE LEGAL CONTEXT OF THE HRGS

Regarding the legal context for housing renovation grants the main provisions are included in the followings Acts, Circulars from the Department of the Environment and Statutory Instruments- Regulations:

Housing Act 1985(Act 1985) makes provisions:

- with respect to Repair Notices (Part VI), Houses in Multiple Occupation (HMO) (Part XI) and (Part VII) as amended by the Act 1989.

Local Government and Housing Act 1989 (Act 1989) make provisions:

- with respect to changes in the definition of unfit properties (Part IX and Schedule 9);
- on dealing with unfit properties (Part VII);
- for house renovation grants (Part VIII).

Circular 6/90, Local Government and Housing Act 1989, Area Renewal, Unfitness, Slum Clearance and Enforcement Action (1990):

- gives guidance on changes with respect to action on unfit properties and clearance areas.

Circular 12/90, Local Government and Housing Act 1989, Housing Renovation Grants (1990):

- gives advice and information on the provisions for a new house renovation grant system;
- explains the provisions for group repair; and
- explains the arrangements for Exchequer contributions.

Circular 4/90, Assistance with Minor works to Dwellings (1990):

- gives advice on minor works assistance.

Circular 10/90, House Adaptations for People with Disabilities:

- deals specifically with provision of disabled facilities.

Circular 5/91, Local Government and Housing Act 1989, Parts VII and VIII (1991):

- gives information on test of resources.

Circular 12/92, Houses in Multiple Occupation (HMO),Guidance to Local Housing Authorities on Standards of Fitness under section 352 of the Act 1985 (1992):

- gives guidance on the standard of fitness of HMO.

Circular 10/92, Housing and Community Care (1992):

- gives guidance on the role of LA in community care policy.

Circular 7/93, Local Government and Housing Act 1989: Changes to parts VII and VIII:

- introduces changes to parts VII and VIII of Act 1989.

S.I. 1990 No. 388, The assistance for Minor Works to Dwellings, Regulations 1990:

- gives guidance for minor works assistance.

S.I. 1990 No. 1189, The Housing Renovation etc. Grants (Reduction of Grant) Regulations 1990:

- gives guidance for the test of resources.

S.I. 1990 No. 1236, The Housing Renovation etc. Grants (Prescribed Forms and Particulars) Regulations 1990

- introduces the forms for grant application.

S.I. 1993 No. 551, The Housing Renovation etc. Grants(Reduction of Grant) (Amendment) Regulations 1993:

- introduces amendments to the guidance for the test of resources.

S.I. 1993 No. 552, The Housing Renovation etc. Grants (Prescribed Forms and Particulars) (Amendment) Regulations 1993 :

- introduces amendments to the forms of application for house renovation grants.

S.I. 1993 No. 554, The assistance for Minor Works to Dwellings (Amendment), Regulations 1993:

- introduces amendments to the minor works assistance.

Together, all of the above mentioned provisions, and others, are intended to form a comprehensive legal framework within which the HRGS operates. The intention behind the new provisions is to secure a reduction in the number of unfit dwelling-houses.

2.2.4- THE NEW FITNESS STANDARD FOR HUMAN HABITATION

A new fitness standard for human habitation was introduced by means of section 83 of schedule 9 of the Act 1989. The new fitness standard provides means for determining whether a property is fit for human habitation, by (Department of the Environment, 1990a):

- a criteria and a list of requirements that must be met if the properties are to be considered fit for human habitation. It applies to both houses and flats;
- providing detailed advice, on the application of the requirements that form the fitness standard; and
- referring to a list of references (Building Regulations, British Standards, Codes of Practice, Drafts for Development or other codes and guidance's relevant to new building work) against which the severity and extent of defects in existing buildings may be assessed.

Under the new fitness standard a dwelling is deemed unfit for human habitation if its condition fails to meet one or more of the requirements listed in the table 2.4.

The importance of the fitness standard is that it is currently the key criterion in determining whether a property is eligible for grant aid repairs and/or improvements. Applications for renovation grants, accompanied by a certificate of owner occupation, must be approved where the works concerned are designed to render a property fit, subject to the provisions that: i) the applicant must be able to comply with the required preliminary conditions; and ii) renovation is the most suitable course of action. Equally, the fitness standard is the criterion by which houses are examined for possible inclusion in clearance programmes.

An objective interpretation of the standard of unfitness is important for a number of reasons:

- to ensure an accurate assessment of the condition of the property;
- to allow fair and equal access to mandatory grants under HRGS; and
- for the just administration of compulsory housing powers.

Table 2.4: Requirements of the fitness standard for human habitation

A dwelling is considered fit if:
- it is free from serious disrepair; - it is structurally stable; - it is free from dampness prejudicial to health of the occupants; - it has adequate provision of lighting, heating and ventilation; - it has an adequate piped supply of wholesome water; - there are satisfactory facilities for the preparation and cooking of food including a sink with a satisfactory supply of hot and cold water; - it has a suitably located WC for exclusive use of occupants; - it has, for the exclusive use of the occupants, a suitably located bath or shower and wash-hand basin, each of which is provided with satisfactory supply of hot and cold water; - it has an effective system for the draining of foul, waste and surface water.

2.2.5- THE IMPACT OF THE FITNESS STANDARD ON HOUSING

The EHCS91 shows a sharp increase in the number of unfit dwellings due to the additional requirements in the new fitness standard. The reasons for the classification of a house as being unfit were reported by the EHCS91 (Department of the Environment, 1991) and are listed in table 2.5. Figure 2.3 below shows the incidence of unfit dwellings by reason for their unfitness.

The most common reasons of unfitness found by the EHCS91 in 1991, are a failure to comply with the requirements of: repair (39%), food preparation (39%), bath/shower/wash hand basin provision (25%), dampness (22%) and water closet provision (19%).

The reasons for unfitness varied with the construction date and tenure. Failure to comply with the requirements of repair, stability and dampness were more prevalent in older stock, pre-1919.

In the stock post-1919 there was a greater incidence of problems associated with internal facilities of the dwelling (food preparation, bath/shower/whb, wc).

Table 2.5: Distribution of unfit dwellings by reason for unfitness reported in 1991 (Department of the Environment, 1991)

Requirements (Reasons for Unfitness)	No. Unfit Dwellings	Unfit Dwellings %
Repair	584,000	39
Food preparation	583,000	39
Bath/shower/whb	370,000	25
Dampness	332,000	22
Water closet	278,000	19
Ventilation	272,000	18
Heating	184,000	12
Structural stability	147,000	10
Drainage	134,000	9
Lighting	121,000	8
Water supply	41,000	3

Source: English House Condition Survey 1991, HMSO

The number of items (requirements) on which a dwelling fails the standard of unfitness was used by the EHCS91 as an indication of the severity of unfitness. The EHCS91 shows that over half of the unfit dwellings in 1991 were unfit on only one item. A further quarter was unfit on just two items. And another quarter were unfit on three or more items. The older dwellings are more likely than that more recent construction to be unfit on more than one requirement.

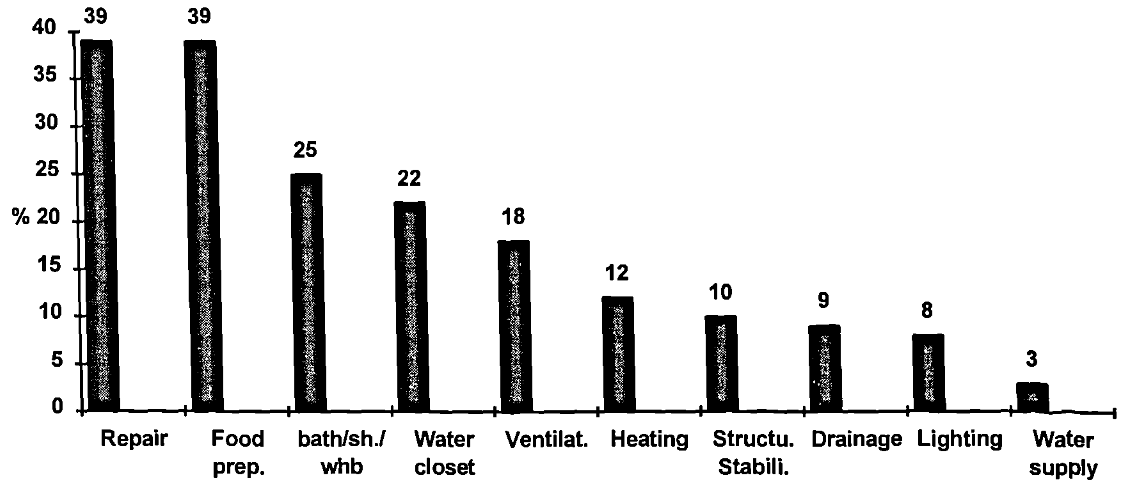


Figure 2.3: Incidence of unfit dwellings and reasons for their unfitness in %

The EHCS91 highlights that unfitness is largely a product of dwelling age and type. Unfit dwellings were found predominantly amongst the older stock, with over half being built before 1919. Pre-1919 terraced houses and converted flats have the highest rate of unfitness. The private sector (owner-occupier and rented) is in by far the worst condition. In absolute terms, the EHCS91 shows that the most unfit dwellings are in the owner-occupied sector. Figure 2.4 below, shows the types of intervention undertaken on unfit dwellings in England, and the predominance of individual renovation grant as a means to bring unfit dwellings into use, over other available actions. This is evidence of the practical importance of the HRGS within the current housing renewal strategy.

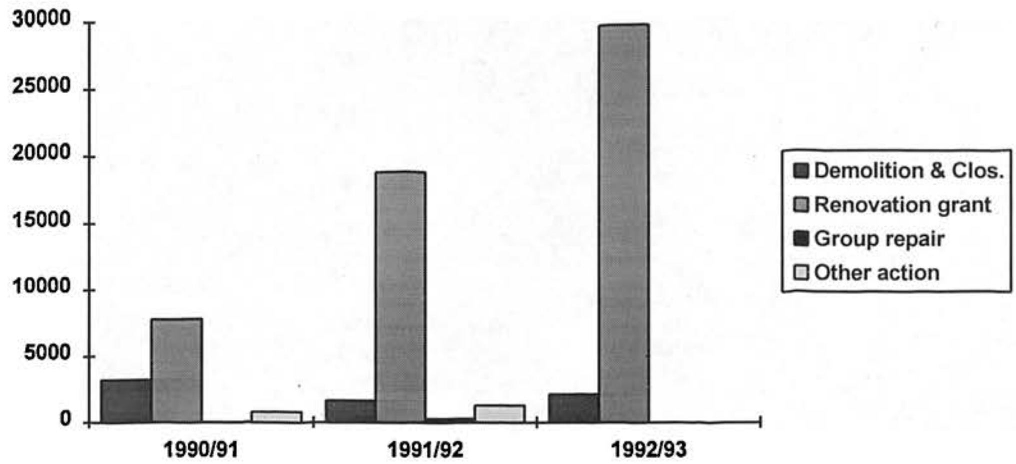


Figure 2.4: Types of intervention on unfit dwellings

2.2.6- THE TEST OF RESOURCES (MEANS TEST)

Within the private housing sector, income is an important determinant in deciding whether or not to give a grant and its amount. According to the EHCS91 (Department of the Environment, 1991), those households in the lowest income band are more likely to live in the poorer condition housing than those in the highest income band.

The amount of a grant is decided by the test of financial resources. There is one test for both owner-occupiers and tenants, and a different test for landlords. The test of financial resources enables the local authority to work out how much the applicant(s) are likely to be able to pay towards the cost of the works. The amount of grant, if any, is the difference between the total cost of the works that are eligible for a grant and the amount that the applicant(s) are likely to be able to afford towards the cost of works.

2.2.7- BENEFICIARIES FROM THE HRGS

Owner-Occupiers:
The owner-occupier proportion of the total English dwelling stock at the end of 1993 accounted for 67.3%. According to the EHCS (Department of the Environment, 1991), the owner occupied-dwellings in poor condition in urban areas accounted for 12% of dwellings. The income of 44% of householders in dwellings of poor condition was less than £6,000 per year. Those in the bottom income band are two and a half times as likely as those in the highest band to live in dwellings of poor condition.

Private Landlords and Tenants:
According to the EHCS91 (Department of the Environment, 1991), 68 % of the private rented dwelling stock was traditional housing (terraced 32.6%; semi-detached 20.8 %; and detached 14.8 %), of which 83 % were built before 1944 (with 67.7% of them built before 1919). The dwellings in poor condition in the private rented sector accounted for 40 % of all private rented dwellings.

Older People:
People aged 75 and over are likely to live in houses lacking amenities. They are nearly one and a half times as likely to live in an unfit house and 1.3 times as likely to live in houses in serious disrepair compared to other households.

Disabled and Infirm People:
Many older people also experience forms of disablement that require major adaptations to improve their ability to live independently or to enable them to remain in their homes. Some younger people also require assistance of this kind.

2.3- BACKGROUND ON GRANT APPLICATIONS

2.3.1- TYPES OF APPLICATIONS

An application for a grant, starts with a preliminary enquiry, followed by a formal application using an appropriate form provided by the LA. There are different forms according to the grant sought and the type of applicant as shown by figure 2.5. Each type of application requires different kinds of information. In general, applications for the renovation, disabled facilities, common parts and HMO grants ask for information about:

- the property and the nature of the works to be carried out;

- the interest of the applicant in the property and how it is being occupied;
- financial resources of the applicant and other occupants; and
- other enclosures like certificate of future occupation.

An application for minor works assistance should include information about:

- the applicant/occupant needs and characteristics; and
- the property and required assistance.

The postal survey mentioned in Section 2.1 has confirmed that the renovation grant is one of the predominant grants. The survey has shown that in 58.6% of the responding LAs the percentage of applications eligible for mandatory grants is between 81-100 %. In about 28% of the responding LAs the percentage of applications for mandatory renovation grants is between 41-80% of all applications.

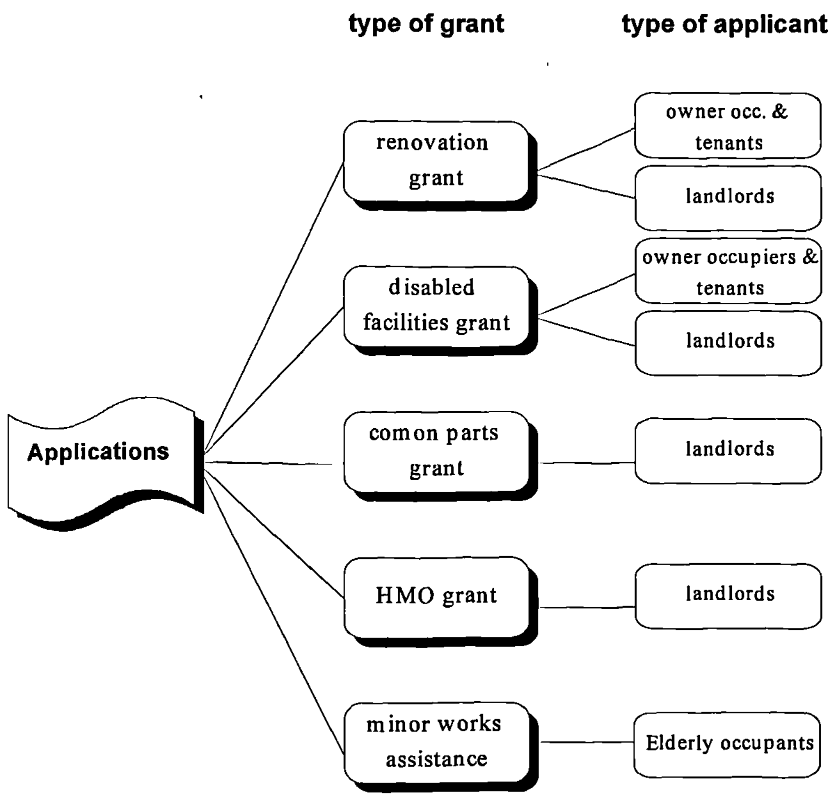


Figure 2.5: Types of applications by the type of grant and tenure of the property

The postal survey has shown that 60 to 80% of applications are for terraced houses built before 1919. Of all grant applications, over 60% are for individual properties located outside any area based scheme, 0 to 20% for properties within a group repair scheme, 0 to 20% properties inside a clearance area and 20 to 30% for properties inside a renewal areas.

In this research an application comprises the preliminary enquiry and the formal application.

2.3.2- SERVICE PROVIDED TO THE APPLICANTS

The postal survey has shown that the types of services provided by the responding LAs to the applicants for the HRGS vary with each authority. 100% of the responding LAs provided the following kinds of services to the applicants (except minor works assistance):

- property survey and assessment;
- the schedule of works (renovation/adaptations);
- test of financial resources or common called by means test;
- assistance in completing an application; and
- supervision during implementation of the works.

Additionally, 62% of the responding LAs provide a list of contractors who have satisfactorily carried out previous grant aided work. Some 90% of local authorities carry out a preliminary enquiry before a formal application is made.

2.3.3- THE ASSESSMENT OF GRANT APPLICATIONS IN PRACTICE

Some 62% of the responding LAs use a particular guide or framework to help with their assessment of grant applications. Each LA bases its work on the legal framework described in section 2.2.3 and its own experience. They adopt different procedures for each grant, that suit their local circumstances and conditions. Few LAs use an informal means test at the time of the preliminary enquiry to assess the eligibility of the applicant/households. At the Client, a team of experienced staff carry out the work to assess the applications and to ensure that the works are carried out in accordance with the grant specifications and Building Regulations. These environmental officers are supervised by an environmental chief officer. The Client has developed its own application forms and guidelines to help with the assessment of the condition of the properties.

Figures 2.6, 2.7 and 2.8 show representations of the general procedures used by the Client for assessing grant applications respectively for renovation and disabled facilities, and minor works assistance. These representations show that the overall task of the assessment of grant applications is a complex activity which consists of a number of interrelated sub-tasks. Each diagram represented by the figures 2.6, 2.7 and 2.8 mirror the entire process followed by each application according to the type of grant sought.

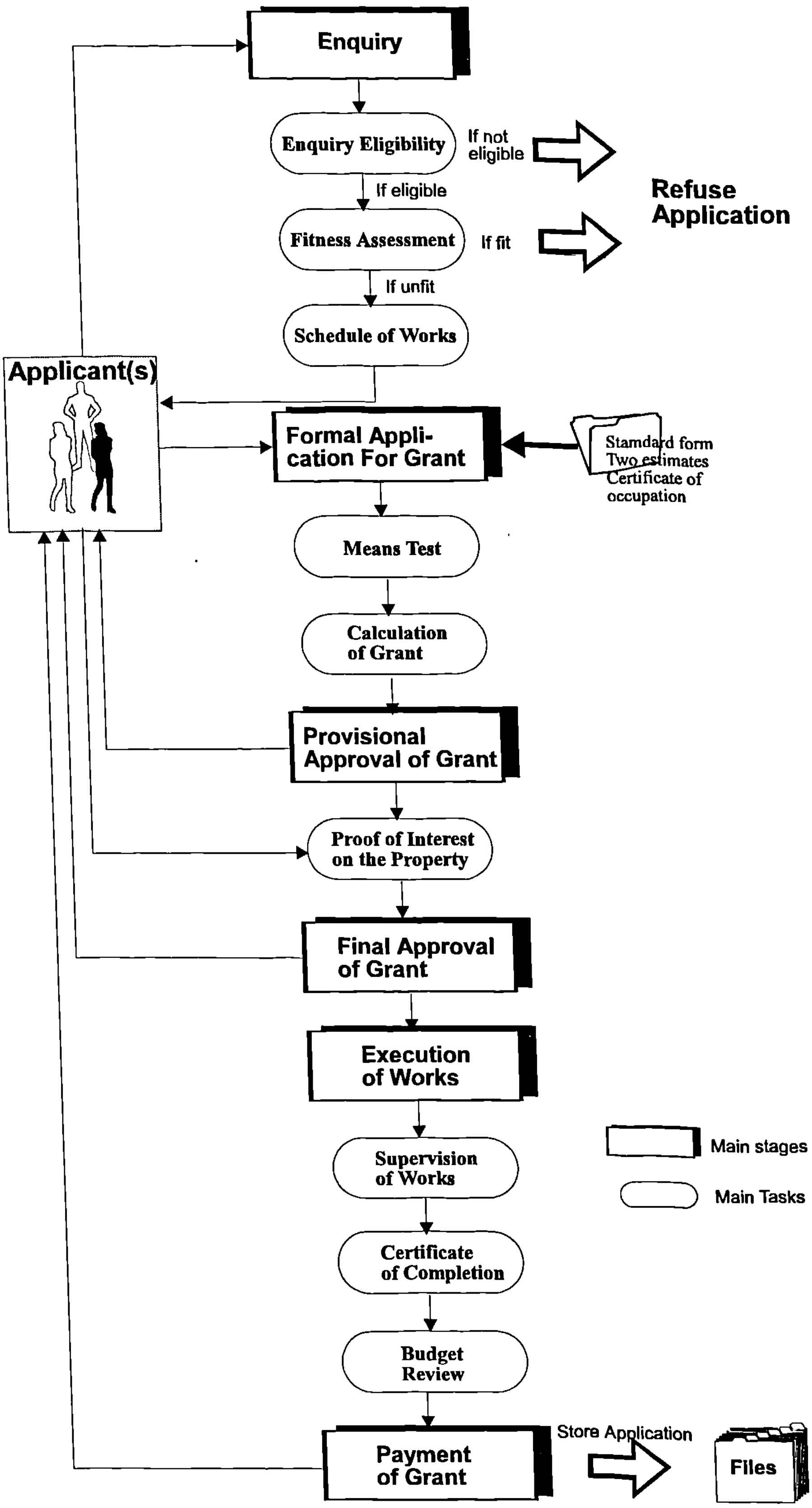


Figure 2.6: General procedure for assessing applications for renovation grant

The above procedure underlies the process followed to assess and implement an application for renovation grant.

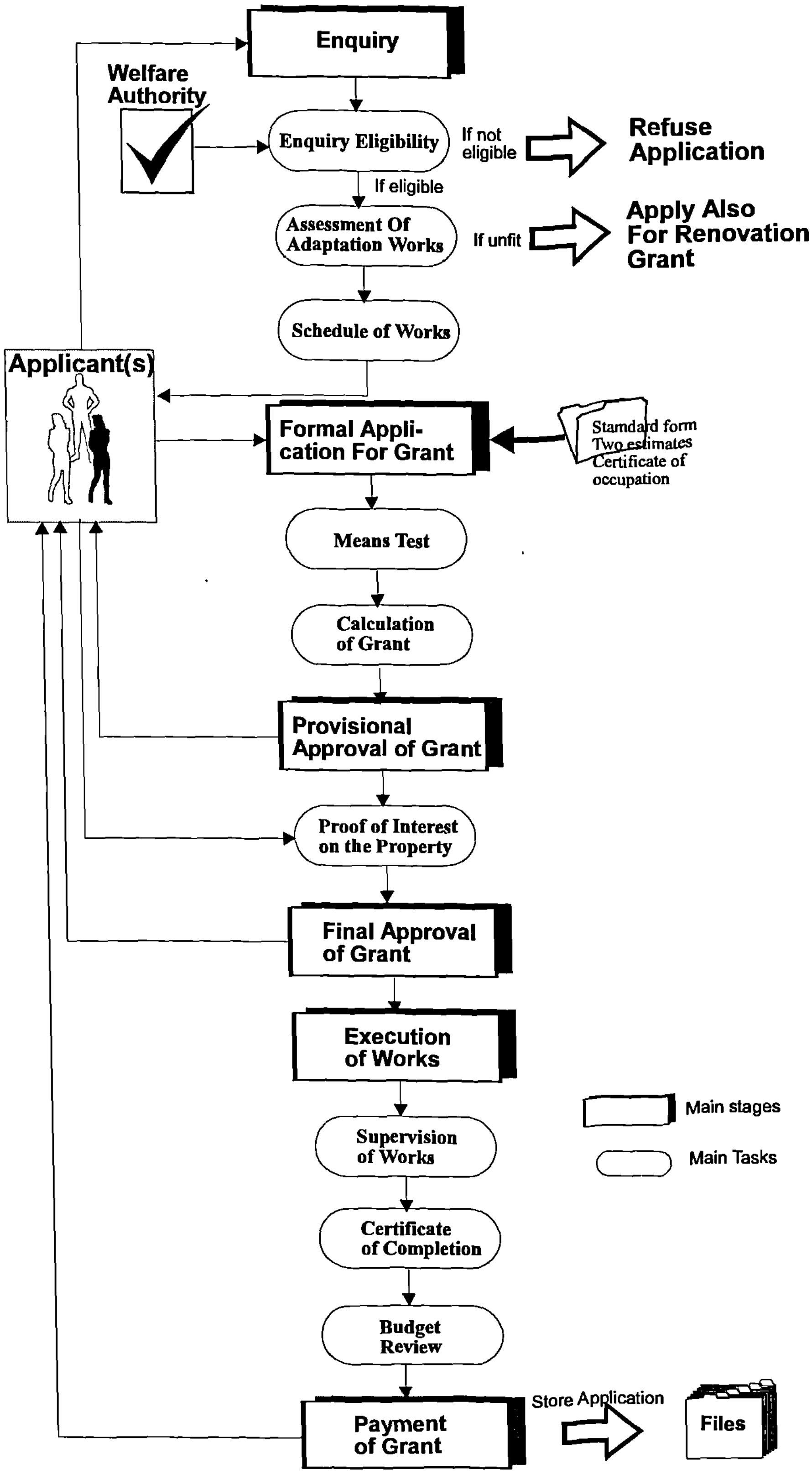


Figure 2.7: General procedure for assessing applications for disabled facilities grants.

The Client was not awarding discretionary grants at the time of the research. According to the Client, any application for a renovation grant from a property that was deemed unfit according to the standard of fitness is automatically considered eligible for grant. The amount is decided by the means test. The above procedure underlies the process followed to assess and implement an application for disabled facilities grant. In the case of applications for disabled facilities, and where the property is unfit, then another application should be made for a renovation grant.

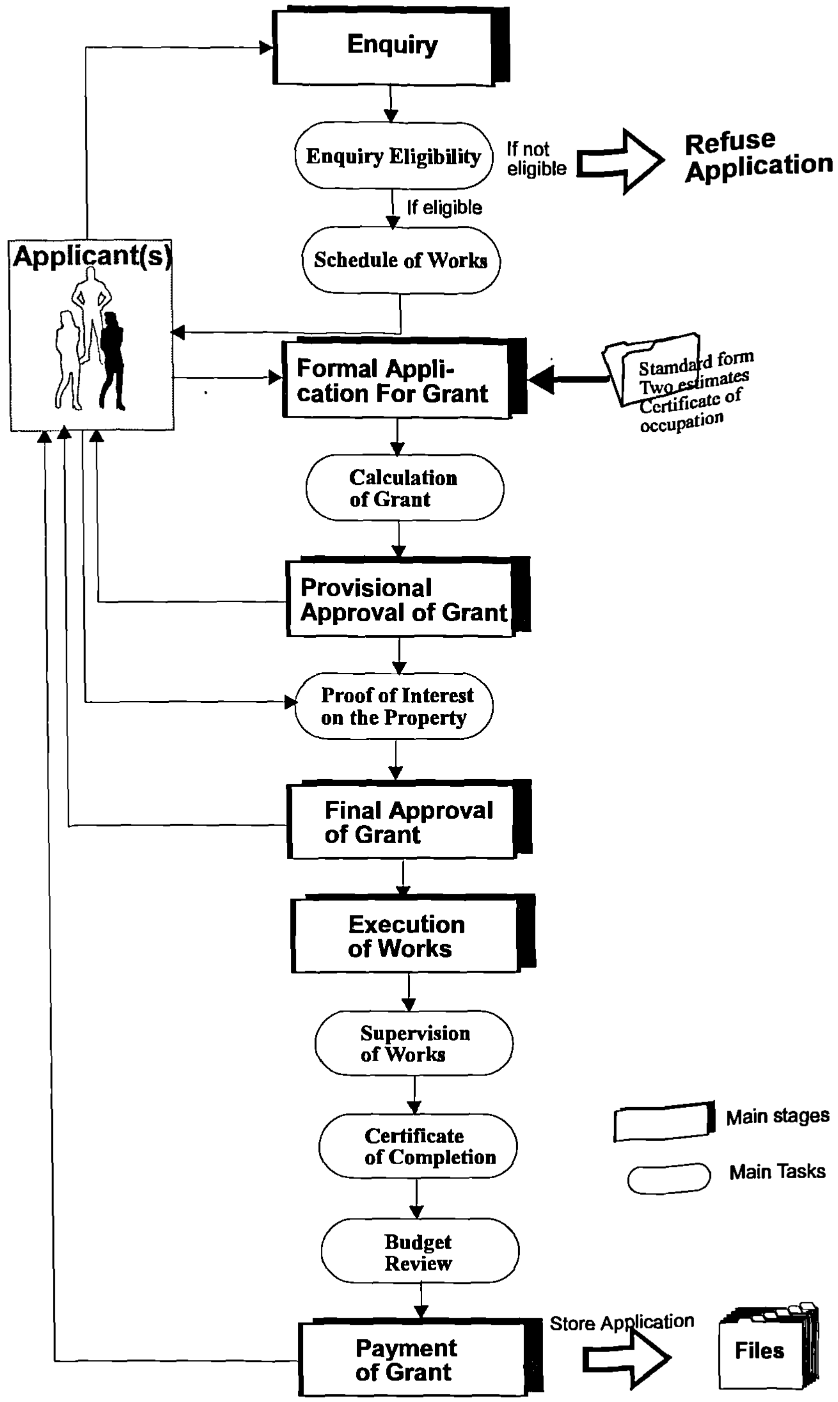


Figure 2.8: General procedure for assessing applications for minor works assistance.

The above procedure underlies the process followed to assess and implement an application for minor works assistance.

The postal survey findings indicated that about 86% of the responding LAs use computers to assist them in performing some of the tasks of the assessment of grant applications. The software used is from a variety of sources. About 40% of the responding LAs are neutral or unhappy with the existing software. The Client uses a network of computers linked to a mainframe to assist them with following tasks:

- recording and managing the applications data;
- calculating the means test; and
- processing payments.

2.3.4- APPLICATIONS BY TYPE OF TENURE

The postal survey has shown that over 70% of all grant applications are from owner occupiers, 20% from private tenants and 10% from landlords. Figure 2.9 shows the distribution of applications by type of applicant.

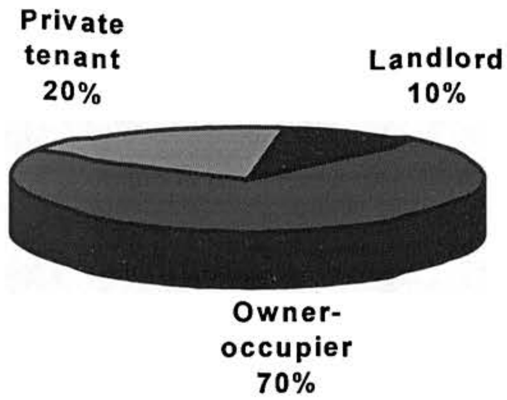


Figure 2.9: Distribution of applications by type of applicant

2.3.5- THE ROLE OF PAST EXPERIENCE

Much of the activity of the assessment of grant applications seems to be guided by the knowledge gained over time through past work. The role of past experience in guiding the assessment of applications for the HRGS can be explained by:

- the continuing changes taking place in the legal framework as a result of the experience gained from the implementation process;

- the way the human expert base their decisions; and
- the guidance notes and standard procedures introduced by each LA.

In the Client organisation there are a number of officers who have vast experience and have processed numerous applications and others with less experience who have processed just a few applications. According to Anderson (Anderson J., 1993), as people become experts in a domain their memory for problems improves because they learn the patterns that appear.

Looking at the role of past experience in the assessment of grant applications the postal survey has shown that some 55% of the responding LAs were of the opinion that they use a lot of the experience gained from past applications to assess new applications. Only 13.8% were of the opinion that they rarely use their past experience in assessing new applications. These results make it clear that experience from past applications might have an important role in the problem solving activity of the assessment of grant applications.

Figure 2.10 shows the distribution of opinions of how LAs use their past experience to assess new applications.

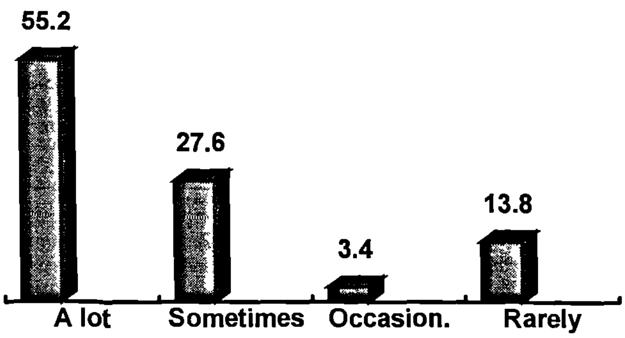


Figure 2.10: How differently LAs opined on how they used experience gained from past application to assess new grant applications (expressed as a percentage)

2.3.6- SIMILARITIES ON GRANT APPLICATION

Out of the responding LAs, 44.8% of them expressed that there are strong similarities or analogies among the grant applications. About 52% expressed that there are some similarities or analogies among grant applications. These results show that there are similarities among grant applications. Figure 2.11 shows the distribution of opinions regarding the similarities found among grant applications.

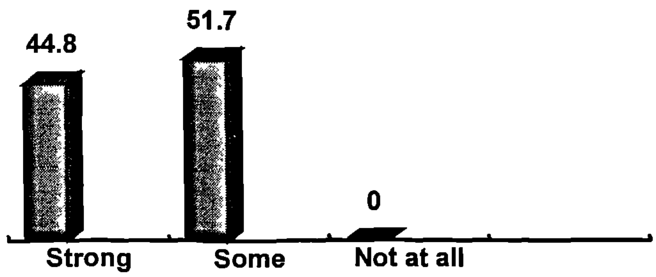


Figure 2.11: How LAs opined about the existence of similarities or analogies among grant applications expressed in percentage

2.3.7- USEFULNESS OF OLD SOLUTIONS

The usefulness of past applications solutions in guiding the problem-solving on new applications was expressed by the responding LAs in following way: 20.7% of them expressed an opinion that solutions contained in the past applications are very useful in guiding the assessment of new applications; 41.4% of them were of the opinion that they are useful; 27.6% of them thought that they are of some use; and 10.3% considered them to be not at all useful. These results show a clear majority in favour of the opinion that solutions from past applications can guide the assessment of new grant applications. Figure 2.12 shows how the councils expressed their opinion.

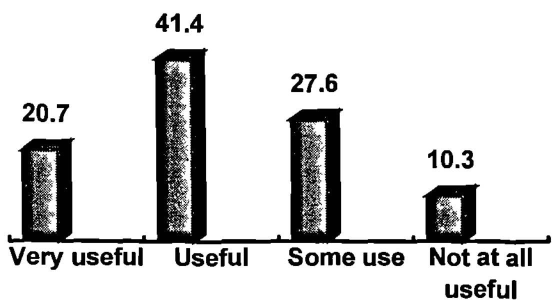


Figure 2.12: How LAs opined on how useful are past solutions in assessing new grant applications expressed in percentages (%)

2.4- SUMMARY AND CONCLUSIONS

In this chapter, the main characteristics of the HRGS domain have been discussed, which provides the background for modelling the proposed system in terms of knowledge that they may contain. From the discussion introduced in the above sections, some conclusions can be formulated as follows:

- The assessment of applications for the HRGS is a complex activity mainly for the following reasons:

    - it is comprised of several interrelated tasks processed in the majority of cases, in a staged fashion;
    - a number of problem-solving strategies are used to accomplish the tasks, each one requiring knowledge from different sources;
    - it is completely data driven activity; and
    - different goals can be established for each task, depending on the type of grant sought, the type of household, the characteristics of the property and local circumstances.

- The differences in the assessment process for different grant types can be explained by the predominance or absence of some of the tasks in the context of the overall task due to the provisions provided by the legal framework and experience.

- Mandatory renovation grants are the most common type of intervention for improving the housing condition in the private sector.

- The condition of the property regarding the fitness standard for human habitation, amongst other aspects, is the key criterion in determining whether or not a property is eligible for a grant. The means test is the key factor in determining the amount of grant to be paid if any.

- The predominance of applications for mandatory renovation grants, mandatory disabled facilities grants and minor works assistance coming from the owner-occupied housing sector is clearly highlighted.

- There are strong indications of the importance of knowledge gained from experience in assessing new grant applications. Solutions that have been used in the past grant applications can be useful, in similar situations, for solving new problems. Past grant applications are the main text records of the relevant knowledge that have been applied previously in practical situations. These past applications are available in large numbers in some LA.

- Similarities can be found amongst grant applications for the same grants, both in terms of the condition of the properties and the households.

- Little work has been carried out to extract knowledge from the information which has been created since the implementation of the HRGS and to make it available in the right format for solving new problems. One of the reasons for this is that current computer approaches are designed to process only specific tasks of the overall task.

In the next chapter, some of the research carried out in CBR will be described, and the main applications developed in general, and in the construction industry so far in particular will be discussed.

CHAPTER 3

KBSs USING CASE-BASED REASONING

3.1- INTRODUCTION

In the previous two chapters the main problems concerning with the assessment of applications for the HRGS and the current situation have been discussed. The lack of effective computer tools to support human experts in all stages of the process of assessing a grant application was highlighted and the application of AI techniques has been stressed as having the potential for improving the current situation.

In this chapter are reviewed some of the most relevant issues of KBSs using CBR with interest to the present research. These issues include: i) case acquisition; ii) knowledge representation; iii) case memory organisation; iv) case indexing and retrieval; v) adaptation; and vi) architectures including CBR. In this context, several KBSs in the research and industrial fields are discussed in some detail, in terms of the problems which they address. These KBSs include those developed in the area of the construction industry and those using CBR combined with other problem solving methods.

The current development of KBSs using CBR and the practical lessons learnt from development and use of a number of existing systems provided support for establishing a set of guidelines for the system developed in this research These guidelines are presented at the end of this chapter.

3.2- BACKGROUND OF CBR

3.2.1- CONCEPT OF CBR

CBR has become popular as a psychological theory of human cognition behaviour which deals with issues such as memory, planning, remembering, problem-solving, decision-making and learning (Kolodner J. and Riesbeck K. 1986; Riesbeck C. and Schank R. 1989; Slade S., 1991; Vargas J. and Raj S., 1993; and Kolodner J., 1993). As a growing technology, it is perceived by a wide community of researchers as computationally attractive for the development of KBSs in a wide range of problems. As stressed by Hammond (Hammond K., 1992) "CBR has grown from a centric view of cognition to a solid sub-area that is supported by wide-spread academic research and industrial development".

In assessing the development of CBR, Kolodner (Kolodner J. and Mark W., 1992) referred to early CBR systems as autonomous problem solving systems, but recent ones work as interactive external memories for users who actually solve the problem.

Several authors have tried to provide a concise definition for CBR. Hammond (Hammond K., 1992) suggested a definition for CBR which looks at:"

- CBR as nothing new.
- CBR as an alternative cognitive model.
- CBR as an approach to knowledge engineering.
- CBR as a new set of assumptions.
- CBR as a new set of modelling goals.
- CBR is part of a larger model distinguished by the view of agents and environments as dynamic entities that change to fit each other over time. This model supports the idea that reasoning and learning must be linked within any intelligent system ".

Leake (Leake D., 1994), defined CBR as: "a method for solving new problems by retrieving and adapting relevant solutions from a memory of past cases". Kolodner (Kolodner J., 1993) provided another definition, stating that: "CBR can mean adapting old solutions to meet new demands, using old cases to explain new situations, using old cases to criticise new solutions, or reasoning from precedents to interpret a new situation or create an equitable solution to a new problem". Along with this definition, Kolodner pointed out that CBR suggests a model of reasoning that incorporates problem solving, understanding, and learning. The majority of researchers have suggested that CBR operates as a theory, a method and a model of reasoning which integrates problem solving and learning.

3.2.2- MAIN AREAS OF RESEARCH ON CBR

Significant ongoing research has focused on some of the key issues of CBR, including (Leake D., 1994): i) *case acquisition*; ii) *case representation*; iii) *case indexing*; iv) *case retrieval*; v) *case adaptation*; vi) *learning*; and vii) *architectures*. The application of CBR for solving real-world tasks has included: i) *planning and scheduling*; ii) *design*; iii) *diagnosis*; iv) *decision aiding*; and v) *teaching and tutoring*.

According to Leake (Leake D., 1994), the NCAI-93 Workshop on CBR, which brought together over 60 investigators from academia and industry, highlighted the success of many systems which have already been deployed, especially as decision aiding systems. The workshop also pointed to *case adaptation* as the least understood aspect of CBR and, consequently, a particularly rich area for research.

Decision aiding, *planning* and *design* are the most popular reasoning tasks to which CBR has being applied at both research and field levels.

The assessment of grant applications has the intention of finding a solution plan of the best course of action for a specific application. A KBS in the domain of the assessment of grant applications can function as a decision aiding tool.

Following the traditional classification of major task areas, a system for the assessment of grant applications might fall within the decision aiding task area.

3.3- KBSs USING CBR ALONE OR COMBINED WITH OTHER PROBLEM SOLVING METHODS

3.3.1- SOME RELEVANT APPLICATIONS

According to Goel, Kolodner, Pearce, Billington and Zimring (Goel A. et al., 1991) designing and implementing a system raises several issues specifically related to the use of CBR methods, such as: i) How cases can be acquired? ii) What information should the cases contain? iii) How should cases be organised in the case library? iv) What methods are used to retrieve cases from the case memory? v) What methods are used to adapt an old solution? One way to study these questions is to analyse how the recent systems have addressed these and other issues (Kolodner J., 1993).

Section 3.2.2, pointed out some of the fundamental issues addressed by the research community in recent years. Thus, for the purpose of development of the system proposed in this research, some of the systems using CBR developed to date were studied and presented according to the way they address the issues mentioned in Section 3.2.2. Through this presentation, it was possible to improve understanding of how some of the relevant implementation issues were studied and solved in practical situations.

Table 3.1 presents some of the representative KBSs using CBR (purely CBR systems and those that integrate CBR with other problem solving methods) in task areas such as: decision aiding, planning and design. These systems are classified according to the implementation issues they address.

Table 3.1: Some of the representative KBSs using CBR alone or with other problem solving methods classified by the implementation issues they address

System	**Domain**	**Task Area**	**Implementation Issues**						
			Case acqui-sition	Know-ledge repres.	Inde-xing	Retrie-val	Adap-tation	Archi-tec-tures	Lear-ning
Abby	Lovelorn probl.	Advice-giving			*****	*****			
ADMS	Mechanical devi.	Diagnosis		*****				*****	
Anon	Planning failures	Advising		*****	*****	*****			
Archie	Architectural design*	Aiding design		*****	*****	*****			
BankXX	Statute of bank-ruptcy	Advising		*****					
Cabaret**	Area of tax law	Law interpretation		*****				*****	
Cabins**	Manufacturing	Scheduling	*****		*****	*****	*****		*****
Cadet	Mechanical devices	Design			*****		*****		
Cadsyn **	Buildings*	Structural design		*****	*****	*****	*****		
Cascade	Drivers of VMS operating system.	Advising		*****		*****			
Casey**	Heart failures	Diagnosis		*****					
Celia	Automobiles	Car diagnosis	*****					*****	*****
Chef	Meals	Planning					*****	*****	
Chiron**	Tax	Lawyer assistant				*****		*****	
Cyclops**	Landscape*	Design						*****	
Clavier	Autoclave load	Design		*****	*****	*****	*****		
Compac	Computer hardw.	Advising support				*****			
Compac** Quick-source	Network printer problems	Diagnosis						*****	

System	Domain	Task Area	Implementation Issues						
			Case acquisition	Knowledge repres.	Indexing	Retrieval	Adaptation	Architectures	Learning
Floabn	House devices	Operation		*****					
Grebe**	Workers' law	Legal advice		*****		*****		*****	
Hypo	Trade law	Legal advice		*****				*****	
Judge	Criminal sentences	Legal advice					*****		
Julia	Meal planning	Design					*****	*****	
Kritik**	Physical devices	Design			*****	*****	*****	*****	
Memorabilia	Architectural design*	Tutoring		*****					
Mediator	Mediation	Planning						*****	
Orca	Organisational change	Advising				*****			
Persuader	Labour negotiation	Labour disputes advising			*****	*****	*****		
Prism**	Interbank telexes	Text classification		*****				*****	
Prodigy	Various	Planning		*****					
Protos	Audiology	Diagnosis				*****			*****
Router**	Path planning	Planning					*****	*****	
Scavenger	Mechanical and electronic device	Case acquisition	*****	*****					
Squad	Software quality control	Advising	*****	*****					
Struple	Architectural Design*	Design		*****	*****				
TaxOps	Tax law	Advising			*****				
VTAEXS**	Technical assistance information in resolving vertical launch systems problems	Diagnosis		*****	*****			*****	

* Systems in the construction industry
* * Systems including CBR with other problem solving approaches

Table 3.1 includes systems developed in the construction industry. In this presentation, systems developed in areas other than the construction industry are also listed. It was decided to extend the study presented in this chapter beyond those developed for the construction industry for three reasons: i) applications in the construction industry do not address all of the issues related to the design and development of systems using CBR; ii) according to Kolodner (Kolodner J., 1993) CBR is still relatively young; and iii) as more systems are studied in this research, more concrete guidelines for building the proposed system will be developed.

In the following sections an attempt is made to identify how the systems listed in table 3.1 have addressed the issues mentioned in Section 3.2.2.

3.3.2- HOW THE SYSTEMS HAVE ADDRESSED THE IMPLEMENTATION ISSUES

3.3.2.1- Case Acquisition

Cabins (Miyashita K. and Sycara K., 1993)
Cabins, designed for scheduling in the manufacturing domain, acquires the users' scheduling preferences in the form of cases. These acquired cases are then reused, to generate and manage schedules in response to unexpected events.

Scavenger (Zito-Wolf R. and Alterman R., 1993)
Scavenger acquires cases from everyday episodes of instructions from a person or a text manual relating to mechanical and electronic devices.

Squad (Kitano H. et al., 1993)
Squad acquires cases following a process with several phases. Each software quality control activity is acquired and modelled as a case.

3.3.2.2- Knowledge Representation

Abby (Domeshek E., 1991)
Cases in Abby contain descriptions of everyday situations (stories of everyday events) which include the most important causal mechanisms at work in these situations and intentional causation. The structure of Abby cases includes the following major group of features: theme; goal; plan; action; and effect. The case structure and the case-base

organisation are packaged into a kind of intentional chain that describes a problem rather than explaining an action or state as being positively motivated by a goal. The underlying structure of Abby cases is a vector of features.

ADMS (Feret M. and Glasgow J., 1993)

Cases in ADMS store a fragment of a past experience, i.e., a past diagnostic scenario, consisting of: i) the fault occurred; ii) the series of pruning steps used to produce a list of potential diagnoses; iii) the list of potential diagnoses produced by the structural isolation process; and iv) the correct diagnosis selected by the operator.

Archie (Goel A. et al., 1991; Pearce M. et al., 1992)

Cases in Archie describe specific building designs. These design cases capture and structure the knowledge that architects remember about specific instances of buildings that they have designed. A design case is specified along three dimensions. The first dimension specifies the goals and constraints of the design (requirements), the design plan (the physical structure of the office building) and the outcome of the design plan (how the design plan satisfied the goals and constraints). The second dimension specifies the characteristics of the client organisation. Finally, the third dimension specifies the structure of the office building. Cases in Archie are represented as flat, static frames, each with more than 150 possible features. Features can be concepts, texts, integers, real numbers, or functions. Most features are concepts and text.

BankXX (Rissland E. et al., 1993)

The case base in BankXX consists of a semantic network whose nodes represent cases and legal theories, and whose labelled links represent connections between nodes. Case-graph nodes are legal cases represented: i) as factual situations; ii) as bundles of citations; iii) as stereotypical stories or scripts; iv) in terms of various legal factors, and v) by the measure of their typicality. Legal theories are also represented as case-graph nodes. The case graph is partitioned into spaces.

Cadsyn (Maher L. and Balachandran B., 1994)

Design cases in Cadsyn are stored as a combination of an object-oriented representation of the design concepts (and the behaviour of the design) and drawings of the design solution (convey the structural appearance of the design). The object-oriented representation serves as the indexing scheme and the drawings provide a visualisation of the design and how it works. Cadsyn's case memory can be viewed as a collection of design cases organised as a set of super-case/sub-case hierarchies. Each design case is broken into a number of sub-cases.

Cascade (Simoudis E. and Miller J., 1991; Simoudis E., 1992)

The knowledge base is a database of cases representing past problem episodes reported by customers. These cases contain records of the past problem solving experience of repairing crashes which have occurred in VAX/VMS operating systems. Each case in the case base is a vector of feature/value pairs organised in an object-like structure. A case in Cascade has three parts: i) the surface features whose values are inexpensive to obtain (which provide information about the failure and the environment in which the crash occurred); ii) the relevant derived features ("validation" contains information that is used during the justification phase) and pointers to the probes that derived them; iii) the repair action (solution) that solved the problem. The case base was organised into a generalisation hierarchy using the UNIMEN algorithm. Because of the type of problem solving that is performed by help desks, the solution of a relevant case is used without modification.

Casey (Koton P., 1988)

Cases in Casey are represented as frames (feature:value type) including three major slots: i) the description of the problem to be solved; ii) the solution; and iii) a justification for the solution.

Chiron (Sanders E., 1991)

Chiron uses three types of knowledge sources: i) prototype plans; ii) previous tax cases; and iii) a set of dimensions. Cases are used not only as a basis for a solution, but also to indicate the boundaries within which a solution can be found. Chiron's prototypes are based partly on the common-sense meaning of the statutory predicates. Additional information can be obtained from the cases. Cases are represented using a structure much like Hypo's legal case frames: both are based on the case summaries.

Clavier (Hennesy D. and Hinkle D., 1991, 1992; Hinkle D. and Toomey C., 1995)

Clavier's cases represent successful configurations of autoclave load schedules and are structured to capture the following information: i) the parts and their relative positions on a table; ii) the tables and their positions in the autoclave; iii) the classification of the layout; and iv) production statistics. Cases in Clavier are useful for generating layouts and load schedules.

Compaq Quicksource (Nguyen T. et al., 1993)

The Compaq Quicksource case base was divided into several partitions: hardware problems, software problems, networking problems and general information. Each case consists of the following information: a title, a description field that describes a case's symptoms in natural language, a question area and a solution section.

Grebe (Branting L. and Porter B., 1991)

Grebe's knowledge base consists of rules and cases. Both rules and precedent cases are necessary in Grebe's domain of worker's compensation law because the most important knowledge sources include legal rules with very abstract antecedent and precedent cases with detailed facts that inevitably differ in many particulars from any new case. Grebe's case library contains published legal precedents concerning the compensation under Texan worker's compensation law for injures sustained while a worker is travelling. In addition, the knowledge base includes paradigm cases representing stereotypical situations.

Hypo (Ashley K. and Rissland E., 1988)

The representation of Hypo's cases concentrates on situation descriptors necessary for the interpretation and argumentation of trade situation problems. User inputs and cases are represented in a simplified form, using a standard "legal case-frame" to hold important facts of a specific case. Legal case frames also include such information as the date of the decision, the court deciding the case, and the official citation. Each of the cases in Hypo's case base is stored using a fixed set of indices.

Memorabilia (Oxman R. E., 1991)

Memorabilia stores architectural design cases in its case library. These design cases capture representative past architectural designs which are made available to students who are learning design. They concentrate on the spatial organisation of architectural designs. The design cases are represented as natural language descriptions linked to appropriate illustrations.

Prism (Goodman M., 1990)

A case in Prism represents pre-classified telexes used to classify inter-bank financial telexes. The cases in Prism's library are organised into a binary decision tree where: i) leaf nodes are clusters of text sharing the same classification; and ii) internal nodes are binary discriminations based on the presence or absence of tokens and classes of tokens in the text.

Router (Goel A. et al., 1994)

A case in Router contains three kinds of information: i) the initial and goal locations in a past planning episode; ii) the spatial neighbourhoods the locations belong to; and iii) the path connecting the two locations.

Scavenger (Zito-Wolf R. and Alterman R., 1993)

Cases in Scavenger represent procedures derived from experiences of the operation of mechanical and electronic devices. Scavenger uses a multi case plan representation of episodic procedural knowledge.

Squad (Kitano H. et al., 1993)

Cases in Squad capture corporate-wide experiences. Each case corresponds to a software quality control activity carried out by company experts. Cases are represented in a structured style of attribute-value pairs and stored in a relational database.

Struple (Zhao F. and Maher M., 1988)

Struple uses a database to store structural design solutions of existing buildings as design cases. The design description in a case includes general information, geometric information, architectural specifications, load information, three-dimensional systems, two-dimensional systems, floor systems, and foundation systems.

VTAEXS (Small R. and Yoshimoto B., 1995)

Cases in VTAEXS reside in the configuration-controlled case-base. Each case includes a set of fields, such as: i) title; ii) problem description; iii) associated questions with weights based on appropriate answers; and iv) repair actions to be taken. Within these fields are embedded pointers to the hypertext document, where related information can be found regarding the theory of operation of vertical launch system and how this case relates to the current technical assistance.

3.3.2.3- Indexing and Retrieval

Abby (Domeshek E., 1991)

The indexing in Abby is based on intentional analysis of a situation. The indices include problems, since a problem is what calls for advice. The indexing scheme is based on the chain-interactions that package a problem with an account of why the problem occurred.

Archie (Goel A. et al., 1991; Pearce M. et al., 1992)

Archie uses an indexing scheme composed of design goals, outcomes and situation descriptors. Archie uses two mechanisms for retrieving cases: i) nearest-neighbour matching and primitive concepts to retrieve building designs that satisfy a problem's goals and constraints; and ii) model-based clustering, which are domain models to cluster cases in the memory. Models in Archie represent domain knowledge about office building design.

Cabins (Miyashita K. and Sycara K., 1993)

Cabin's indexing scheme is based on manufacturing domain relations and constraints. Cabin has a retrieval algorithm based on the serial search of memory and nearest-neighbour matching.

Cadsyn (Maher L. and Balachandran B., 1994)

In Cadsyn indexing data is organised as a scheme in which the relation, function, behaviour, and geometry of a design case are explicitly represented. Cadsyn's indexes are simply based on relation, function, behaviour and structure attributes. This indexing mechanism allows the system to find designs that have similar features in these four categories. The design projects, or parts of design projects, are retrieved not only on the basis of the structure attributes, but also on relations, functions, or behaviour attributes through a hierarchical search. The function and structure indexes are used to explore the case memory in order to find relevant cases. The case memory is organised into two components: i) case hierarchy; and ii) case indexing representation.

Cascade (Simoudis E. and Miller J., 1991; Simoudis E., 1992)

Each case in the case base is a vector of feature-value pairs, where indexes are based on the available descriptive features of the current problem. Cascade's retrieval algorithms perform according to the following steps: i) to retrieve those cases which appear to be the most relevant, based on the available descriptive features; ii) for each retrieved case, to derive features that can validate the usefulness of the case; iii) to use the newly derived features retrieve additional cases; and iv) to continue returning cases that match well.

Clavier (Hennesy D. and Hinkle D., 1991, 1992; Hinkle D. and Toomey C., 1995)

Given a list of parts, Clavier searches its library for previously successful configurations (cases) that: i) minimise the number of parts that are not on the list; ii) maximise the number of high-priority parts; and iii) maximise the total number of parts. Clavier's cases are indexed by parts they include. Thus, Clavier uses the input (a list of composite parts) to retrieve cases, by choosing as the best case the one that includes the most parts of highest priority.

Compaq Smart (Acorn T. and Walden S., 1992)

Cases in Compaq Smart use an indexing scheme based on a checklist of hardware features. The user describes the problem, and the system retrieves cases by conducting a serial search.

<u>Grebe (Branting L. and Porter B., 1991)</u>

Grebe's cases are indexed by the goals they achieve and the structural characteristics of snippets. Grebe uses a structural, semantic network representation of case facts. The facts of a retrieved past case are compared to those of a new case by the mechanism of structure matching.

<u>Kritik (Goel A., 1989)</u>

Kritik cases are indexed by the functions delivered by the stored designs. In Kritik, the functional specification of the desired mechanical design is matched with the functional specifications of stored designs.

<u>Persuader (Sycara K., 1987)</u>

Persuader's cases are indexed by their predictive features in labour negotiations. A user describes a labour dispute to the system, it creates a solution using the best precedent-setting case it can find. If necessary, the system augments that solution with pieces of solutions from other cases.

<u>Router (Goel A. et al., 1994)</u>

Cases in Router are indexed by the initial and goal locations of the stored plan, and by the spatial neighbourhoods the locations belong to. In case retrieval, Router uses the output of the neighbourhood-find task as a probe into the case memory to search for cases that match the current problem as closely as possible. In particular, it searches the neighbourhoods containing the two locations, first looking for cases exactly matching the specified task, then for partial matches.

<u>Scavenger (Zito-Wolf R. and Alterman R., 1993)</u>

Cases in Scavenger are indexed by an indexing vocabulary that includes: goals they achieve; plans being carried out; and steps in the plan. Retrieval in the case library is carried out by best-match on primarily first order features.

<u>Struple (Zhao F. and Maher M., 1988)</u>

Given a description of the new building, Struple finds relevant past structural solutions in the case library and uses the information stored in past solutions to plan the structural configuration of the new building. Matching is done using a similar metric that compares significant common aspects of the matched buildings and the current building. The similar metric is described by a set of matching criteria, which is a set of requirements on some specific features.

__VTAEXS (Small R. and Yoshimoto B., 1995)__

VTAXES uses character matching algorithm provided by ART-IM shell. The authors claim that character matching is the most robust of the three text-matching algorithm provided by ART. Once the new case data have been entered, the user goes to the Search Case-Base screen provided by CBR Express. Based on the user's entry of the problem description, VTAEXS uses ART-IM text-matching algorithm to find candidate cases in the case-base that have some degree of match.

3.3.2.4- Case Adaptation

__Cabins (Miyashita K. and Sycara K., 1993)__

Cabin uses the violations of constraints to point to the need for adaptation of a retrieved case.

__Chef (Hammond K. 1986,1989)__

Chef uses one of the substitution methods available for case adaptation. It uses the reinstantiation method to create a chicken and peas recipe from a recipe for beef and broccoli. Thus, chicken is substituted for beef everywhere in the recipe, and peas are substituted for broccoli.

__Clavier (Hennesy D. and Hinkle D., 1991, 1992; Hinkle D. and Toomey C., 1995)__

Clavier performs adaptations using the case-substitution method. Clavier's case-based substitution method uses pieces of existing cases to suggest substitutions. Clavier's initial retrieval phase sometimes retrieves cases that have unmatched parts and, hence, the system needs to find a substitute for the unmatched part. The system searches for possible compatible substitutions by looking for pieces of cases with a similar context. Clavier suggests part substitutions to the user and provides the cases used to determine the parts' compatibility. Then the user can choose among these, or any other available part on the list, to replace the unmatched parts.

__Judge (Bain W., 1989)__

The Judge system uses parameter adjustment as a substitution method to sentence a criminal to a shorter sentence than another criminal because the crime was less violent. The parameter adjustment of an old criminal sentence relies on specialised heuristics that relate differences in input specifications to differences in output.

Julia (Hinrichs T. and Kolodner J., 1991)

Adaptation in Julia is viewed as switching to a smaller, more tractable search space. It uses specialised adaptation heuristics to modify the structure of its solutions. Thus, adaptation is a kind of heuristic search in which transformations are applied to a source concept in order to repair constraint violations.

Kritik (Goel A., 1989)

Kritik implements model-guided repair to perform design adaptations of a past design candidate. It has incorporated some repair heuristics that address the mechanisms behind the modifications of a past design. To perform the design adaptation, Kritik takes the behavioural specification of a desired design and a set of past design candidates (ordered by their ease of adaptation) and finds the structural specification of a modified design that satisfies the behavioural specification of the desired design.

Persuader (Sycara K., 1987)

Persuader uses parameter adjustment as a method of adaptation. Thus, the parameter adjustment method is used to adjust the numerical values of an old contract to bring them into line with a new case. Persuader uses heuristics to adjust the parameters of an old contract. Some of these heuristics adjust a set of parameters all at once, based on the differences between the old and new situations, while others are specialised to particular parameters.

Plexus (Alterman R., 1986)

Plexus uses a local search as a method for making substitutions to a retrieved past case plan. Plexus has incorporated abstraction hierarchies that represent the relationships between goals and plans, where the search for substitutions takes place.

Router (Goel A. et al., 1994)

Router uses a recursive processing strategy for adapting a path plan. It formulates path-planning sub-problems, recursively spawns new path-finding sub task, finds the solutions to the new path-findings sub-problems, and combines their solutions with the initially retrieved route.

3.3.2.5- Architectures

Systems reviewed in this chapter have used various types of architectures. They are: i) purely CBR architectures; ii) architectures that incorporate other methods in its support of

CBR; iii) architectures in which CBR methods are applied in conjunction with other methods; and iv) architectures where CBR comes in support of other methods.

Archie (Goel A. et al., 1991; Pearce M. et al., 1992)

Archie uses a CRB architecture augmented with domain models to cluster design cases in memory.

Cabaret (Rissland E. et al., 1991)

Cabaret mixes reasoning paradigms in interpretative reasoning in support of argumentation. More specifically, Cabaret incorporates independent rules and cases in combination to construct arguments with regard to the various concepts involved under the Internal Revenue Code. Rules are used to perform statutory interpretations. Cabaret incorporates a control strategy that decides how and when it is appropriate to invoke rule-based reasoning and CBR.

Cabins (Miyashita K. and Sycara K., 1993)

Cabins integrates CBR with the constraint propagation method for scheduling.

Cadsyn (Maher L. and Balachandran B., 1994)

Cadsyn combines CBR and the decomposition method to support structural design synthesis.

Cyclops(Navinchandra D., 1992)

Cyclops uses an architecture that combines CBR with rule-based reasoning. It moves from one to another of its reasoners until it finds one that solve its problems.

Casey(Koton P., 1988)

Casey uses an architecture where CBR comes in support of other problem solving methods.

Chef (Hammond K. 1986, 1989)

Chef employs a purely CBR architecture, i.e., it uses only the CBR method.

Chiron (Sanders E., 1991)

Chiron integrates prototypes, cases and rules to interpret the current problem and construct a plan for tax planning in the context of multiple past cases.

Compaq Quicksource (Nguyen T. et al., 1993)

Compaq Quicksource uses rule-based reasoning coupled with CBR search engine. Rule-based reasoning is used to pre-answer as many questions as possible based on the user's problem description. Rules can also answer question based on previously answered questions. CBR is used to search cases in the case base.

Grebe (Branting L. and Porter B., 1991)

Grebe incorporates a problem solving model that uses complementary of rules and precedent cases in the classification task, in the domain of Texan worker's compensation law.

Julia (Hinrichs T. and Kolodner J., 1991)

Julia uses an architecture that incorporates other methods in support of CBR method.

Kritik (Goel A., 1989)

Kritik integrates CBR, model-based reasoning and other methods into a single framework for design problem solving. Kritik incorporates a control strategy that is provided by the task structure for the design problem.

Prism (Goodman M., 1990)

Prism combines CBR with rule-based reasoning for text classification and routing.

Router (Goel A. et al., 1994)

Router's architecture integrates model-based and case-based reasoning for performing the path planing task. The model based-reasoning provides navigational plans.

VTAEXS (Small R. and Yoshimoto B., 1995)

VTAEXS architecture includes case-based and model-based reasoning. VTAEXS principally uses CBR to match current cases with a library of historical and canonical cases. Model-based reasoning supports user understanding of technical issues implied by the CBR and analysis of problems not currently addressed by the case-base. Model-based reasoning is not tightly coupled to the CBR. The use of CBR was justified because records of more than 500 historical tech-assist cases over the life of the vertical launch systems (VLS) were available.

3.3.3- DISCUSSION OF SYSTEMS USING CBR IN THE CONSTRUCTION INDUSTRY

In contrast to the relatively large number of applications of CBR in various domains, throughout the literature, there is little research work or few CBR systems reported in the construction industry. Most of the systems using CBR in the construction industry were developed for design tasks, such as: architectural design and tutoring, structural design and landscape design. Although CBR has proved to be useful for a wide range of tasks such as: planning, diagnosis, advising and legal reasoning.

CBR programming environments (ARCHIE, used ReMind shell) and programming languages (CADSYN) were used for the development of some of the systems listed in table 3.1. Most systems are still at research level with the aim of testing the utility of CBR in specific tasks such as design. Issues such as, case acquisition, case adaptation, and case validation are not addressed in detail. It seems that CBR developments from other areas can be applied creatively in addressing tasks related to the construction industry.

Most systems use an object-like representation (ARCHIE, CADSYN) or natural-language representation (MEMORABILIA) for case representation. The object-like representation describes each case by a set of features and their values. These features can be concepts, text, numbers and functions. Features are generally grouped into the main case's components, such as: i) problem/situation specification; ii) solution for the specified problem; and iii) outcome of the solution. Cases represent individual designs.

A hierarchical organisation for the case library was adopted in most of the systems (ARCHIE and CADSYN).

Indexes are generally part of the case representation vocabulary and they are context related. They are a subset of the case representation. Simple similarity metrics are used in the matching procedures (STRUPLE and CADSYN). Case retrieval mechanisms in some systems are provided by the development tool (ARCHIE) or have been developed using programming languages (STRUPLE and CADSYN).

There are systems which are purely case-based (ARCHIE and MEMORABILIA) and others which include CBR combined with other methods (CADSYN and CYCLOPS). A characteristic common to all systems is that none of them is aimed at replacing human experts completely, but supporting them in some of the design tasks with an external memory.

3.4- GUIDELINES FOR THE PROPOSED SYSTEM

The review of the main systems using CBR provided some useful guidelines for establishing the main features of an architecture suited for the system proposed in this research and for its consequent implementation.

Some of the systems listed in table 3.1 used considerable resources in hardware and software for their development. One of the main limitations for the development of this system was concerned with hardware and software. The limited amount of resources available for the research discarded the use of CBR programming environments and powerful hardware. Another restriction was related to the time available. A number of limitations had to be established regarding implementation issues, the type of problems and domain areas the system will address. Taking into account the review of the main systems, the limitations of this research and the characterisation of the domain discussed in chapter 2, a number of concrete guidelines for the development of the proposed system were formulated as follows:

- A micro-computer based tool provided with object-oriented programming tools and text-matching algorithms should be suitable for the development of the system proposed in this research.
- The characterisation of the domain discussed in chapter 2 shows that the overall task of the assessment of grant applications has several tasks. CBR might be appropriate for some tasks of the overall task but not appropriate for others. Therefore, the review of presented in section 3.3.2 shows that there are a number of systems which combine successfully CBR with other methods. Thus, it is convenient to determine: i) what are the sub tasks which the CBR method can decompose the overall task; ii) which of those tasks will be the responsibility of the CBR; iii) what kind of methods are needed for remaining tasks; and iv) how CBR will interact with other methods to accomplish the overall task.
- An important component to be addressed in the context of the proposed system should be the design and implementation of the case library. Issues related to the acquisition, representation and indexing of cases must have careful attention.
- A hierarchical organisation for the case library seems to be suitable for the domain. The review of CBR systems shows that a number successful systems, such as CLAVIER, use a hierarchical organisation for the case library.
- It seems that past application cases stored in the case library may not provide solutions for all new problems. Thus, the proposed system should be provided with the ability to perform some kind of adaptation. A number of reviewed systems have successfully used

substitution methods for adaptation of past solutions. Application cases can be used to suggest adaptations in the proposed system.

- The proposed system should be designed to provide support to the human expert. It should act as a decision aiding support system for people that possess some knowledge about the HRGS. The system should allow a certain level of interaction with the user and provide explanations for its reasoning activities.
- The validation of the case library should be carried out from the skeleton stage. Initially the case library should be established with a set of valid seed cases, which must be selected from a representative sample. Thus, the system will be able to generate reasonably accurate results from the early development stages.

Finally, taking into account the restriction of time, the development of the application should attain the working system stage of the CCA method. At this stage of development, a system is reasonably validated and in theory can be used in practical situations (Brandon et al., 1988). At the working system stage it will also be possible to validate the system using field tests.

3.5- SUMMARY AND CONCLUSIONS

In the first part of this chapter the concept of CBR and its main areas of research were discussed. In the second part, some of the most important systems are reviewed in the context of these areas of research. This review aimed to establish relevant guidelines for the proposed system which are presented in the third part of this chapter. These guidelines are based on the lessons and experiences learned from other studies, the nature of the domain of the HRGS and the limitations of this research in terms of resources and time.

Although CBR has been applied to a wide range of tasks, such as design, planning, legal reasoning, advising, diagnosis and tutoring, and its usefulness has been tested, the number of systems in the construction industry so far is very limited.

The review shows that CBR as a method for building KBSs offers a lot of flexibility which allows the developer to use his or her creativity and imagination to make it work. There are systems which are purely case-based. Others have different methods to support CBR. Other systems combine CBR with other methods in a similar way. Therefore, some systems have emphasised CBR issues more deeply than others. According to Kolodner (Kolodner J., 1993), concrete rules for building CBR systems are still not as readily available as they are for the rule-based systems. For this reason, even using the most simple CBR approaches and limited resources, it will always be possible to learn something from the development of an application in a practical domain where there is knowledge that can be modelled in terms of cases.

The system proposed in this research will be a micro-computer based decision aiding support system, aimed at modelling the knowledge available to assess grant applications. A commercial shell will be used to implement the system following the CCA method.

The following chapter provides an analysis of the system's task in terms of: i) the sub tasks which the CBR method decomposes the overall task; ii) for which tasks the CBR method is responsible; and iii) the other methods required to perform the remaining tasks.

CHAPTER 4

ANALYSIS OF THE TASK OF ASSESSING APPLICATIONS FOR THE HRGS

4.1- INTRODUCTION

A general characterisation of the HRGS domain has been outlined in chapter 2. This characterisation included a description of the procedures followed by the Client for assessing grant applications respectively for, renovation, disabled facilities and minor works grants. These procedures show that the overall task is a complex activity consisting of several interrelated sub-tasks and is completely data driven.

In chapter 3 some of the most relevant issues of applications using CBR of interest to the present research are reviewed. This review has shown that CBR combined with other techniques (rule-based and model-based reasoning) to build KBSs offers a lot of flexibility which allows the developer to use his or her creativity to make it work in a number of real-world tasks.

This chapter outlines the analysis of the overall task of assessing applications for the HRGS in the context of the characterisation of the domain presented in chapter 2. According to Kolodner (Kolodner J., 1993), when building a KBS for a problem solving task consisting of several sub-tasks, the overall task needs to be analysed before the system can be implemented.

In this chapter, firstly some of the current task analysis methods are introduced. Secondly, the selected method is briefly outlined. Finally, in the third part of this chapter the task analysis for the assessment of grant applications is discussed. The task analysis produced three versions of the task structure for the assessment of applications for the HRGS. These versions of the task structure are presented at the end of this chapter.

4.2- SELECTED METHOD FOR TASK ANALYSIS

4.2.1- SOME CURRENT METHODS FOR TASK ANALYSIS

Brandon (Brandon P. et al., 1988) pointed out: "in building a KBS, it is important and necessary to analyse the nature of the problem task in the following terms of: i) the goal(s) of the program; ii) how is the task performed; and iii) what basic strategies are used to perform the task". According to Brandon, the analysis of the problem task provides a basis for: i) formulating a model of the problem solving processes; and ii) selecting a tool to meet system requirements". Kolodner (Kolodner J., 1993) pointed out three main ways to analyse the problem task. They are:

1. *Task analysis*, which involves: i) breaking the overall tasks into sub-tasks; ii) assessing which sub-tasks CBR is the appropriate method for; and iii) assessing other possible methods for remaining sub-tasks.
2. *Generic CBR tasks*, which involves examining which generic CBR tasks are good at performing the various sub-tasks.
3. *Functions cases can profitably fulfil*, which involves identifying for which task cases can provide a basis for making decisions.

A generic CBR task specifies a family of task instances of a certain type. Some examples of generic CBR tasks include: i) case indexing; ii) case retrieval; and iii) case adaptation.

In recent years a number of researchers in Artificial Intelligence (AI) have been advocating that KBSs should be modelled at an appropriate level with respect to a goal or task using tasks, problem-solving methods, domain models and knowledge-roles as mediating concepts (Newell A., 1982; Newell A. et al., 1991; Newell A. and Chandrasekaran B., 1993; Brandon P. et al., 1988; Clancey W., 1985, 1993; McDermott J., 1988, 1993; Steels L., 1990; Chandrasekaran B. et al., 1992; O'Hara K. and Shadbolt N., 1993; Wielinga B. et al., 1993; Smith J. and Johnson T., 1993; and Strosnider J. and Paul C., 1994). As a consequence of this work, several knowledge modelling methods for describing KBSs with respect to a goal or task have been developed to date. Some of these methods provide means to carry out the analysis of the overall task of assessing applications for the HRGS (*tasks analysis* and *generic CBR tasks analysis*) as proposed by Kolodner (Kolodner J., 1993).

Table 4.1 below summarises some of the major current knowledge modelling methods classified by the type of presupposed model and its main outputs. A detailed description of these methods is presented in Appendix 4. This description is based on the main features of each method.

Table 4.1 includes those methods which share the view that knowledge must be modelled with respect to a goal or task. These methods have a number of common features as well as differences. The task structure analysis from Chandrasekaran, Johnson and Smith (Chandrasekaran B. et al., 1992) consists of a unified framework for task analysis which takes into account the similarities and differences of other task-based methods.

The implementation of the system proposed in this research can be greatly enhanced by carrying out an analysis of the overall task of assessing applications for the HRGS using task-based analysis methods. A great number of the knowledge modelling methods listed in table 4.1 claim that the ability to build a KBS is greatly enhanced by describing the system in terms of its knowledge content before its implementation. This, in turn, will have significant effects upon the specification, design, implementation and maintenance of the system (Steels L., 1990; Basden A. et al., 1991; Chandrasekaran B. et al., 1992; Watson I. et al. 1992b; Yost G., 1993; Wielinga B. et al., 1993; and Smith J. and Johnson T., 1993).

The main objectives of the task analysis of the assessment of applications for the HRGS, in the context of the characterisation of the domain are as follows:

- To describe the proposed system in terms of its knowledge content without being tied to the details of its implementation.
- To produce a detailed specification of the proposed system.

The specification of the system resulting from the overall task analysis can provide a basis for:

- designing and implementing an architecture for the system;
- identifying which sub-tasks CBR should be responsible for;
- identifying which kind of methods will have responsibilities for the remaining sub-tasks;
- describing how each method accomplishes the task for which it is responsible for;
- describing how CBR and other methods interact to accomplish the overall task; and
- supporting the evaluation of the system.

Table 4.1: Some major task analysis methods

Method	Presupposed Model	Main Outputs
Heuristic Classification (Clancey W., 1985, 1993)	Problem solving structures	Tasks, metarules and kinds of knowledge
Inference Nets (Brandon P. et al., 1988; Giarratano J. and Riley G., 1994)	Taxonomy of knowledge	Decision nodes, Relationships between nodes and Inference types
KADS methodology (Wielinga B. et al. 1993, Schreiber G., et al. 1993)	Multiple generic models Four-layer framework for knowledge modelling	Task model, Model of expertise, Conceptual model and Design model
Role-Limiting Methods (McDermott J., 1988, 1993)	Problem-solving methods for generic problem classes	Problem-solving methods for the tasks and Roles of knowledge
Generic Tasks (Chandrasekaran B.,1986a, 1986b, 1990)	Generic tasks	Generic task, Problem solving methods and Knowledge types
Componential Framework (Steels L.,1990)	Task-based model, Analysis of the components of expertise at two levels	Task structure, Task typology, Problem-solving methods and Domain models
Problem Solving approach (Puerta A. et al. ,1992)	Task-based model	Task structure (Task , methods, sub-tasks and mechanism)
Knowledge-Level (KLCM) (Smith J. & Johnson T. , 1993)	Task-based computational model	Tasks environment, actions which can be used to perform the task, knowledge to select actions
Knowledge-Level Primitives (O'Hara K. & Shadbolt N., 1992)	Knowledge-level primitives	Structure of the knowledge-level primitives and Problem solving methods
Task Structure Analysis (Chandrasekaran et al., 1992)	Task structure	Task structure (Tree of tasks, problem-solving methods, sub-tasks and knowledge types required by the methods)

In order to carry out the task analysis of the assessment of applications for the HRGS, the task structure analysis method proposed by Chandrasekaran, Johnson and Smith (Chandrasekaran B. et al., 1992) was adopted in this research. Some of the reasons for choosing this method are:

- it provides an analysis of the overall task in terms of its goals, sub-tasks, associated problem solving methods needed to accomplish them and the knowledge required to use the methods. It allows the author to identify which sub-tasks the CBR is more appropriate for and which other methods are appropriate to accomplish the remaining tasks;

- it allows the decomposition of the overall task into more manageable chunks of sub-tasks, using the generic CBR tasks suggested by Kolodner;
- it allows the author to identify where application cases are useful for problem solving; and
- it provides a specification for the system proposed in this research.

4.2.2- THE TASK STRUCTURE ANALYSIS

A general description of the task structure analysis methods is presented in this section. This description is useful for understanding how the analysis of the system's task was carried out in the context of this research.

According to Chandrasekaran, Johnson and Smith (Chandrasekaran B. et al., 1992) the task structure analysis produces a *task structure*. The task structure consists of a tree of tasks, alternative problem solving methods, and sub-tasks applied recursively until tasks are reached that are in some sense performed directly using available knowledge.

Chandrasekaran and his group assumed that, in general, a task can be accomplished using any one of several alternative problem solving methods. A method can set up sub-tasks, which themselves can be accomplished by various alternative problem solving methods. Thus, the task structure analysis associates tasks with problem solving methods that accomplish them.

Chandrasekaran, Johnson and Smith (Chandrasekaran B. et al., 1992) has proposed a task structure for the diagnosis and design tasks. A partial view of the task structure diagram for the design task is shown in figure 4.1. In the diagram the circles represent tasks and the rectangles represent alternative methods.

According to the Chandrasekaran's unified framework, the task structure consists of three main components. These components are:

Tasks: Tasks are specified as transforming an initial problem state with certain features to a goal state with certain additional features. Thus, a task in the task structure is described by its initial and goal states. Chandrasekaran's method makes a distinction between a *task* and a *task instance*. A *task instance* is a particular problem to be solved. In contrast, a *task* specifies a family of task instances of a certain type. For example, the assessment of the fitness of a dwelling-house is an instance of the task of assessing the condition of buildings, which itself is a subclass of the general diagnosis task.

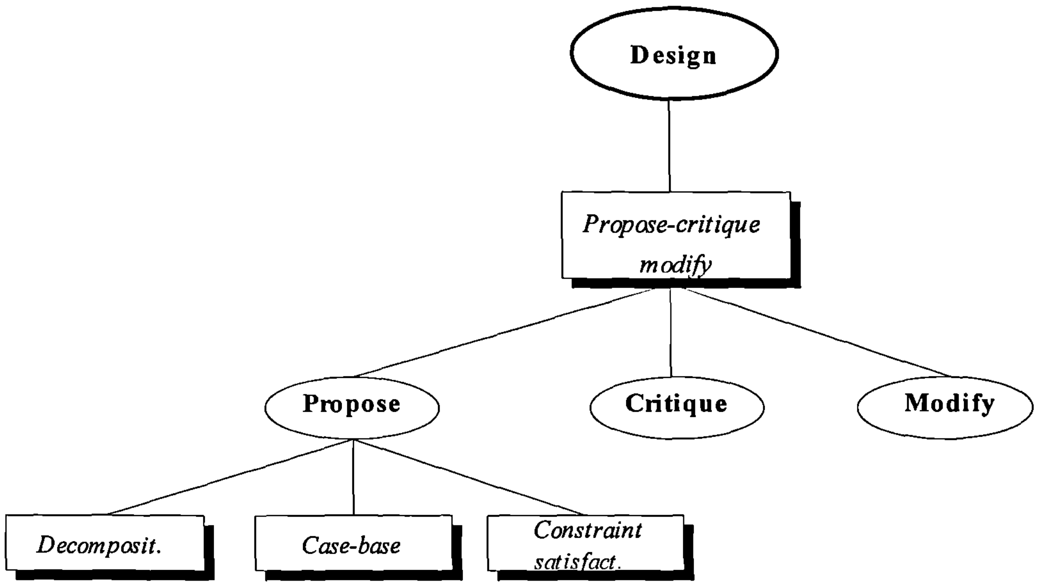


Figure 4.1: Part of a task structure for design. (From Chandrasekaran B. 1990, Chandrasekaran B. et al., 1992)

Problem solving methods and sub-tasks: Problem solving methods in the task structure are regarded as ways of accomplishing tasks and may be of many types. Chandrasekaran (Chandrasekaran B., 1990) has classified the problem solving methods into two types:

- Methods that can be viewed as a search for a solution in a problem space of available solutions. Examples of these methods include: CBR, decomposition, depth-first and best-first methods.
- Methods that consist of algorithms which directly produce a solution without any search in a space of alternative solutions. Examples of this methods include: the numeric and algorithmic methods.

Newell (Newell A., 1982) defined the first type of methods as a search for a goal in a problem space of alternative solutions, where a *problem space* is defined as a set of states and a set of operators linking one state with the next. Figure 4.2 shows a general view of a problem solving method according to Newell's problem space hypothesis.

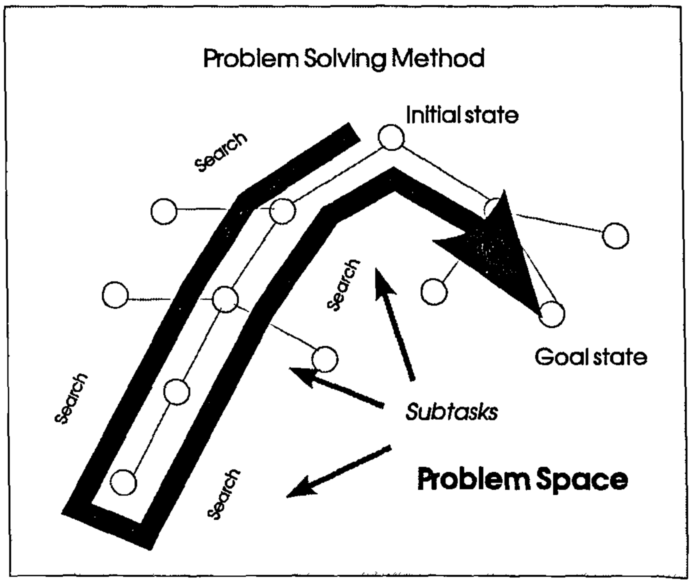


Figure 4.2: Problem solving method search space according to Newell (Newell A., 1982)

Brown and Chandrasekaran (Brown D. and Chandrasekaran B., 1989), based on Newell's problem space hypothesis, assumed that each problem solving method in the task structure is described as a problem space defined by an initial state, a goal state, and some number of intermediate states connected via sub-tasks. Thus, a problem solving method in the task structure is specified in terms of: i) problem space where the search takes place; and ii) a set of sub-tasks that can be used to transform the initial state of a task to the goal state.

Required Knowledge: The knowledge required by problem solving methods in the task structure comes in four different types:

- *Task knowledge.* It is the knowledge needed to accomplish each task which maps the input of the task to its output (goal state). This knowledge alone can lead to the solution of the task by searching exhaustively through the problem space until the goal state is found.
- *Sub-task proposal knowledge.* It is the knowledge used to indicate when a sub-task is needed to be applied to the current state;
- *Search control knowledge.* It guides the search through the problem space by sequencing the sub-tasks which lead to a goal state. Thus, it specifies which sub-task to take from a given state, directing the search for the goal state.
- *Knowledge needed to select a method.* When there are alternative problem-solving methods available to perform the task.

These four types of knowledge in the task structure for assessing applications for HRGS can be available in three forms: i) directly available in the system; ii) derived from the existing knowledge; or iii) provided by the user.

4.3- ANALYSIS OF THE ASSESSMENT OF APPLICATIONS FOR THE HRGS

The task of assessing applications for the HRGS has been analysed in the context of the characterisation of the domain by using the task structure analysis method introduced in section 4.2.2. Thus, task structure analysis in the HRGS domain was assumed to be:

> *a process of decomposition of tasks and problem-solving methods applied recursively until they result in a collection of sub-tasks that can be performed directly using the knowledge and primitive inferences available in the system.*

As a consequence of the task structure analysis carried out, three versions of the task structure were developed corresponding to domain problems which are part of the scope of this research. These versions are:

- Task structure for assessing applications for renovation grants.
- Task structure for assessing applications for disabled facilities grants.
- Task structure for assessing applications for minor works assistance.

These versions of the task structure represent descriptions of the system in terms of its knowledge in three domain problems of the HRGS. These descriptions specify what the system should do. Therefore, they specify how the system should accomplish its tasks in terms of problem solving methods. Each problem solving method is described in the task structure as a search in a problem space of alternative solutions for the current task.

The description of each version of the task structure for the assessment of applications for HRGS consists of three main components:

Tasks: A task is specified by: i) the task definition; and ii) its initial and goal states.

Problem solving methods (way to accomplish a task)**:** A problem solving method is specified by: i) a problem space where the search for the solution takes place; and ii) the sub-tasks in which the task is decomposed by the method.

Knowledge (needed by problem solving methods to guide the search)**:** This knowledge is specified by: i) sub-task proposal knowledge; and ii) search control knowledge. The knowledge required to solve each task is discussed in chapter 5. The three versions of the task structure resulting from this study provided a basis for guiding the acquisition of this knowledge.

4.3.1- DEFINITION OF THE TASK OF ASSESSING APPLICATIONS FOR THE HRGS

The general information processing task of assessing applications for the HRGS can be defined by:

- A set of requirements specified by the current legal framework introduced in chapter 2.
- A technology which consists of a body of available actions for the improvement of the house condition up to a standard of human habitation.
- A set of constraints specific to: i) the grant application regarding the applicant(s), other occupants and houses; and ii) the local policy.

A general description of the version of the task structure for assessing applications for renovation grants is presented in the next sections.

4.3.2- TASK STRUCTURE FOR ASSESSING APPLICATIONS FOR RENOVATION GRANTS

4.3.2.1- Initial And Goal States of The Overall Task

The initial state specifies the input data to the system from a given application for a renovation grant. The goal state specifies what the system should achieve at the end of the problem solving session of assessing that application for a renovation grant.

The initial state for the overall task of assessing an application for the HRGS consists of a set of data contained both in the initial enquiry and application forms. This set of data includes information about: i) the type of grant sought by the applicant; ii) the property and the purpose of the works to be carried out; iii) the applicant's interest in the property; iv) how the property is currently occupied; v) the financial resources of the applicant(s) and other occupants; vi) the needs of applicant(s) and other occupants; vii) the local circumstances regarding the property and its relation to the area; and viii) LA policy.

The goal state is a solution plan containing sufficient information about the applicant, other occupants (if any) and the renovation or adaptation works, so that the house concerned can be improved or adapted up to the required standard of fitness for human habitation.

According to the provisions of the legal framework and taking into account the Client's current practice, a solution plan for a given application for a renovation grant should include the following information:

1. General information about the applicant, other occupants and property.
2. A fitness assessment report about the condition of the property, including: i) its fitness, and if deemed unfit; ii) the reasons for unfitness; iii) the grounds of unfitness and defective building components.
3. A detailed specification of the schedules of work to be carried out in order to repair or adapt the property to the required fitness standard.
4. The budget of works including: i) all costs required to carry out the renovation works; ii) services and charges; and iii) administration fees.
5. An economic appraisal justifying that the renovation action is the most appropriate action to deal with the unfit house.
6. The test of resources of the applicant and other relevant occupants.
7. The degree of eligibility of the application.
8. The amount of grant that can be awarded in order to carry out the works, with and regards to the ability of the applicant to contribute to the cost of works.
9. A decision as to whether or not to award a grant.
10. Conditions of implementation of the grant.

4.3.2.2- Problem Solving Methods

The task structure associates tasks with problem solving methods that accomplish them. Methods specify how the system should accomplish the tasks required for assessing a given application for the HRGS. In general, a task might be accomplished by any of several alternative methods. The task structure must explicitly identify a problem solving method for each task.

Five possible types of problem solving methods were identified for accomplishing the different tasks of the assessment of application for the HRGS. They include: i) CBR method; ii) decomposition method; iii) associative method; iv) abductive assembly method; and v) algorithms. To help assess the appropriateness of each of these problem solving methods to accomplish the different tasks of the assessment of applications for the HRGS, an evaluation of how and why they are applied was carried out. This evaluation is briefly presented in the following paragraphs.

4.3.2.2.1- CBR Method

The basic premise in CBR is that it is more efficient to solve a new problem by starting with a similar old solution stored in the memory, rather than by rerunning all reasoning that was necessary the first time (Kolodner J., 1993). Figure 4.3 shows the classic reasoning cycle of CBR methods and its main reasoning processes. A key aspect highlighted by figure 4.3 is that, in problem solving tasks the CBR method can shortcut the search space by starting by proposing old solutions to new problems, alleviating the need for long and expensive searches required to construct solutions from scratch. To achieve this, CBR uses specific knowledge represented as cases. Past application cases in the HRGS domain hold specific knowledge which can be used to provide almost-right solutions to some problems arising from new grant applications. They are records of how the domain rules (statutory, regulatory and technical) and individual skills (from human experts) were applied in specific situations regarding the households and property. Therefore, past application cases record solutions that have worked well in solving previous problems.

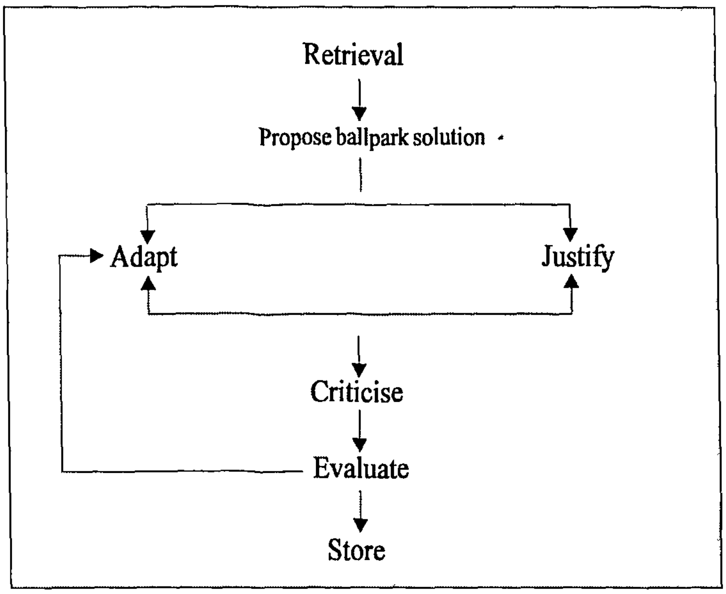


Figure 4.3: CBR reasoning cycle (from Kolodner J., 1993)

In the assessment of applications for the HRGS domain, CBR can perform three major functions by taking advantage of the large body of knowledge held by past application cases:

- to provide almost-right solutions to some of the tasks of the overall task; or
- to provide similar solutions that are then modified to fit the specifications of the new problem; and
- to provide the Client's experts with an external shared memory of past application cases which allows them to share and use their experiences.

By using structured application cases as one of the system's knowledge sources, and deriving solutions based on the information contained in those cases to solve specific tasks, one can take advantage of some of the best features of other problem solving methods.

CBR decomposes a given task into a number of generic tasks of its reasoning cycle. Each of these generic tasks has a search space smaller and simpler than the initial task. According to Kolodner (Kolodner J., 1993) and Goel (Goel A. et al., 1994) the major generic reasoning tasks of the CBR inference cycle include:

- Indexing a new case, which involves the selection of indexes.
- Constructing solutions by retrieving cases from the case library, which involves searching, matching and ranking sub processes.
- Adapting cases or solutions retrieved from the case library, which may involve other sub processes such as, retrieving adaptation types and substituting case components.
- Evaluating cases and solutions retrieved from the case library and cases which have been previously adapted.
- Criticising new cases using cases retrieved from the case library.
- Anticipating failures by using cases stored in the case library.
- Presenting cases for supporting specific tasks such as, reporting and interpreting.
- Storing cases in the case library.

To decide upon the appropriateness of those generic CBR tasks, several questions had to be answered. They include:

- Where is the complexity in the task to be accomplished?
- How much is known in advance about the problem situation?
- Are there many obscure ways in which a solution could fail, or is a reasonable-looking solution almost always a good one?

According to the characterisation of the domain discussed in chapter 2 and the interviews carried out by the author with Client's experts, the more complex reasoning processes involved in the assessment of applications for renovation grants include:

- deriving the schedules of work based on the assessment of the fitness of the property; and
- deriving a decision for a given application.

The information required to derive schedules of work for a given unfit dwelling-house and the solution plan for the application will be known at execution time. This information is not known at the beginning of the problem solving session. Past application cases can provide a basis for specifying schedules of work and for deriving a decision for a given application.

4.3.2.2.2- Decomposition Method

According to Maher (Maher M., 1990) decomposition means that something is decomposed; it also implies a recomposition. Maher (Maher M., 1990) pointed out a number of issues that should be addressed when applying the decomposition method to perform a task. These issues include: What is decomposed? How is the task problem decomposed? Is the decomposition fixed? How does recomposition occur?

Regarding the first issue, the analysis of the assessment of applications for the HRGS should provide an indication of what task problem needs to be decomposed.

How is a task problem decomposed? One way of applying the decomposition method to the assessment of grant applications is by decomposing a specific task into nearly independent or loose sub-tasks in the context of the HRGS domain. The decomposition method helps reduce the size of the search spaces, because the knowledge it uses can be viewed as the compilation or chunking of earlier search in the assessment of applications space. The knowledge needed is of the form $A \Rightarrow A_1, A_2, \ldots, A_n$, where A is a given task, and Ais are sub-tasks associated with search spaces smaller than the original task A (Chandrasekaran B., 1990). The knowledge needed by the decomposition method in the HRGS domain can be provided by the characterisation carried out and described in chapter 2.

Is the decomposition fixed? The characterisation of the domain has shown that for some task problems the decomposition is largely invariant. For example, the task of testing the resources of the relevant persons to a given application is performed by decomposing the test of resources into a number of sub-tasks specified by the legal framework.

How does recomposition occur? According to Chandrasekaran (Chandrasekaran B., 1990) the decomposition method might accomplish a given task by:

- Firstly, decomposing the overall task problem (A) into sub-problems (A_1, A_2,..., A_n).
- Secondly, generating specifications for each sub-problem. The specification of the overall task problem (A) need to be translated into specifications for each of the sub problems (A_1, A_2,..., A_n).
- Thirdly, solving each sub-problem using an appropriate problem solving method and following the order determined by the decomposition of the initial problem.
- Finally, gluing the sub-problem solutions into a solution for the original problem.

For example, the decomposition method accomplishes the task of testing the resources of the relevant persons to a specific grant application by: i) decomposing the test of resources into sub-tasks (finding the relevant persons, calculating the applicable amount, and calculating the eligible income); ii) generating specifications for each sub-task resulted from the decomposition; iii) solving each sub-task in order determined by the legal framework; iv) recompose the individual solutions into solution of the initial tasks, i.e. finding the reduction of grant for that application.

4.3.2.2.3- Associative Method

The associative method uses compiled forms of domain knowledge that can be represented as *rules* and uses *pattern matching* as type of inference.

In a KBS using an associative method, each fragment of knowledge is represented by an *if-then* rule so that whenever a description of the problem situation precisely matches the rule's antecedent (*if* condition), the system performs the action described by this rule's *then* consequent. A *rule* specifies an action, solution or a conclusion to be taken whenever a specific data pattern appears. The user specifies the data pattern and the system inference searches its knowledge base to see it it can find that pattern.

For example, the knowledge required to find the applicable amount with respect to a given application can be encoded as rules of the following type:

if
the relevant person is a single person *and*
the relevant person is aged not less than 25 years
then
the personal allowance is £ 44.00

The *if* parts of the above rule specify the data pattern (condition). The *then* part specifies the personal allowance with respect to such relevant person (solution). Using the above rule the associative method produces a solution by: i) matching the specification of the given relevant person with the specifications of the stored personal allowances (*if* part of the rules of the rule set); and ii) adding this new solution to the knowledge base. Thus, the associative method in the HRGS domain assumes the availability of a store of complete solutions which can be represented in form of *if then* rules.

Associative methods are suitable for: i) domains which require optimal solutions; and ii) performing data-driven tasks. A number of KBSs based on *rules* have been highly successful in solving problems in many well circumscribed domains (Kolodner J., 1991). Rules are small, but consistent pieces of domain knowledge which can be extracted from human experts and/or texts. However, the associative method is inflexible and is only applicable in narrow domains. Also, as the number of rules increases, the search space tends to become large, making the search more expensive (Goel A., 1989).

Due to the nature of the HRGS domain, the associative method can be applied to accomplish those tasks which: i) are data-driven; and ii) the knowledge available can be formulated in terms of *rules*. Some of these tasks include:

- The enquiry eligibility.
- The economic analysis.
- The test of resources (means test).
- The degree of eligibility of a given grant application.

4.3.2.2.4- Abductive Assembly Method

Some researchers (Hamscher W., 1991; Chandrasekaran et al., 1992; Allemang D., 1994; Benjamins R. and Jansweijer W., 1994) working in knowledge modelling pointed out that the general diagnosis can be viewed as an abductive task, i.e., the construction of a best explanation (one or more disorders) to explain a set of data (manifestations or symptoms). Thus, given a set of data S to be explained, a hypothesis H can be found as the best explanation of S when the following criteria is satisfied (Allemang D., 1994):

1. H explains S.
2. Sufficient alternatives of H have been considered
3. The data S is reliable.
4. H is a priori plausible.
5. H surpasses these alternatives by sufficient amount (discrimination).

The abductive methods to diagnosis have at least one advantage over methods in which diagnoses need to be consistent with symptoms: they are more restrictive (Hamscher W., 1991). There are a number of abductive methods which can be applied to the diagnosis task: bayesian, abductive assembly and parsimonious covering. These methods accomplish the diagnosis task by decomposing it into sets of sub-tasks and problem solving methods (Chandrasekaran et al., 1992). These methods are recursively decomposed until primitive inferences are found. According to Chandrasekaran, Johnson and Smith (Chandrasekaran et al., 1992), the abductive assembly method requires knowledge of disorders and the manifestations they explain.

The assessment of the condition of a dwelling-house as required by the HRGS can be viewed as a task instance of the building diagnosis task, which is itself a subclass of the general diagnosis task. The knowledge required by the abductive method to accomplish the assessment of a dwelling-house task is available in the HRGS domain. Thus, the abductive assembly method has the potential to be used to accomplish some of the tasks of the assessment of applications for the HRGS. These tasks include:

- the assessment of the fitness of a dwelling-house with respect to the fitness standard for human habitation in case of applications for renovation grant; and
- the assessment of a dwelling-house with respect to the adaptations required by a disabled person.

These tasks can be viewed as instances of the diagnosis of buildings task. The knowledge required by the abductive assembly method to accomplish both tasks can be extracted from Client's experts and past applications.

4.3.2.2.5- Algorithmic Methods

Algorithms are suitable for so-called well-structured problems. They directly produce a solution without any search in a space of alternatives. For example, in the HRGS domain algorithms can be used to perform the task of: i) finding the amount of grant with respect to a given application; and ii) presenting an application case.

4.3.2.3- Top Part OF The Task Structure

At highest level the task analysis in the context of the characterisation of the domain presented in chapter 2 set up the following tasks:

1. To generate a new application case for the current application with an appropriate structure depending on the type of grant sought by the applicant.
2. To evaluate the eligibility of the enquiry for a grant application.
3. To undertake the assessment of the fitness of the property regarding the fitness standard for human habitation (in the case of an application for a renovation grant) or assessment if the adaptation works are necessary and practicable (in the case of an application for a disabled facilities grant).
4. To determine the schedules of work (renovation, adaptation and minor works).
5. To determine the cost of schedules of work.
6. To undertake an economic analysis of works versus demolition, demolition with redevelopment or maintaining the current state.
7. To undertake the test of resources of the applicant and households.
8. To calculate the amount of the grant.
9. To find the degree of eligibility of the application.
10. To find the decision/solution for the current application.

These tasks can be accomplished by a number of alternative problem solving methods. The task structure analysis allowed the author to identify a problem solving method for each of these sub-tasks. Part of this discussion is outlined in the next sections of this chapter.
Figure 4.4 below shows the top part of the task structure for the assessment of applications for a renovation grant, where the circles represent tasks. Each one of these tasks has a problem space which is smaller than the overall task. These problem spaces are illustrated in figure 4.5.

To illustrate how the task structure analysis was carried out for each of those tasks mentioned above, a description of four of them is presented in the next sections. These tasks are: i) to undertake the assessment of the fitness of the property; ii) to determine the schedules of work for a renovation grants; iii) to undertake the test of resources of the applicant and households; and iv) to find the decision for the current application. A description of the remaining tasks is presented in Appendix 5.

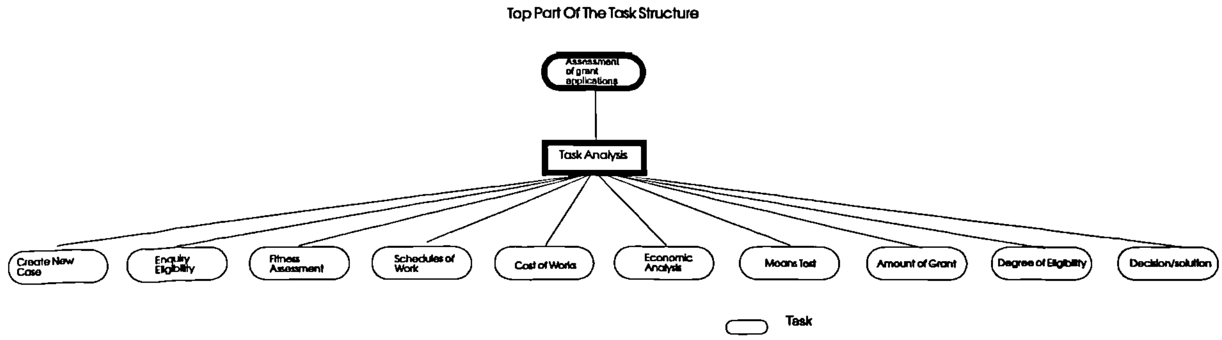


Figure 4.4: Top part of the task structure for the overall task

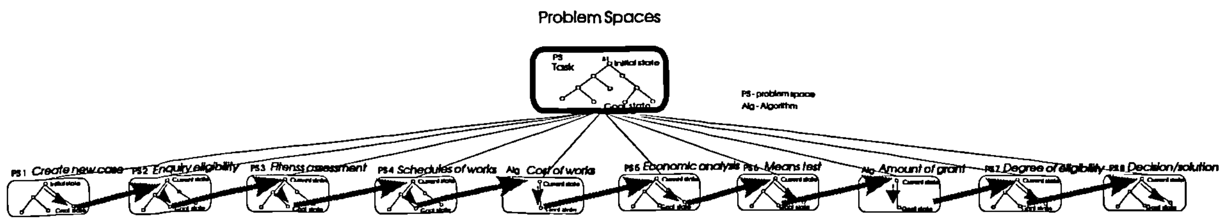


Figure 4.5: Problem spaces to the sub-tasks set by task analysis

4.3.2.4- Description Of The Selected Tasks

4.3.2.4.1- Assessment Of The Fitness Of The Property

Definition: The assessment of fitness task is specified by: i) the observed condition of the dwelling-house; and ii) the requirements of the fitness standard for human habitation.

Initial state: The initial state includes the observations of the condition of the building components with regard to the requirements of the fitness standard.

Goal state: The goal state is the one which includes information about: i) the level of fitness; and if the property is deemed unfit: ii) the reasons for unfitness, i.e. the requirements from the fitness standard that the property failed to satisfy; iii) grounds of unfitness by each requirement; iv) the defective building components; and v) the location of the failure.

Problem-solving method: Chandrasekaran, Johnson and Smith (Chandrasekaran B. et al., 1992) identified the abductive assembly method for the diagnosis task. The assessment of the condition of a dwelling-house can be viewed as an instance of the diagnosis of buildings. Therefore, the knowledge required by the abductive assembly method is available from human experts and texts. Thus, the abductive assembly method was chosen.

Sub-tasks: The abductive assembly method decomposes the fitness assessment task into four sub-tasks: i) *to find evidence of failures*; ii) *to generate hypotheses of the grounds of unfitness;* iii) *to select ground of unfitness; and* iv) *to calculate the level of fitness.*

Sub-task proposal knowledge: Apply the sub-task *to find evidence of failures*, when the inspection of the property takes place. Apply the sub-task *to generate hypotheses of the grounds of unfitness* when the failures to the fitness standard are known. Apply the sub-task *to select ground of unfitness* when the hypothesis of unfitness is known. Apply sub-task *to calculate the level of fitness* when all requirements of the fitness standard have been checked.

Search control knowledge: The abductive assembly method accomplishes the assessment of the fitness of the dwelling-house by: i) finding evidences of failures to the fitness standard; ii) matching the failures to the possible hypotheses of unfitness; iii) selecting the ground of unfitness; and iv) computing the fitness of the dwelling-house.

Domain models: Two domain models were used: the model of building components and the model of the fitness standard.
A description of each sub-task mentioned above is given below.

4.3.2.4.1.1- To Find Evidence Of Failures

Initial state: The initial state includes the observations about the condition of the building components.

Goal state: The goal state includes information about the failures.

Problem Solving Method: The depth-first method was chosen.

Sub-task proposal knowledge: Apply the task when pre-conditions are met.

Search control knowledge: The diagnostic knowledge about the fitness standard knowledge is organised into a classificatory hierarchy. The depth-first method performs the task by searching through all nodes of this hierarchy and ruling-out or confirming evidence of failures.

4.3.2.4.1.2- To Generate Hypotheses Of Grounds Of Unfitness

Initial state: The initial state includes the information about the failures to the fitness standard.

Goal state: The goal state includes the hypotheses of grounds of unfitness which might explain the failures.

Problem Solving Method: The associative method was chosen because this task can be viewed as a selection problem and the required knowledge can be formulated in terms of rules.

Sub-task proposal knowledge: Apply the task when the failures are known.

Search control knowledge: The associative method performs the task by matching the specification of a failure with the specification of hypotheses of unfitness stored in the system.

4.3.2.4.1.3- To Select Ground Of Unfitness

Initial state: The initial state includes the hypotheses of unfitness for each building component.

Goal state: The goal state includes the grounds of unfitness that fully explains the failures to the requirements of the fitness standard.

Problem Solving Method: The associative method was chosen because this task can be viewed as a selection problem and the required knowledge can be formulated in terms of rules.

Sub-task proposal knowledge: Apply the task when the hypotheses of unfitness are known.

Search control knowledge: The associative method selects grounds of unfitness from the hypotheses of unfitness.

4.3.2.4.1.4- To Calculate The Level Of Fitness

Initial state: The initial state is the one which includes all information about the grounds of unfitness.

Goal state: The goal state includes information about: i) the fitness of the dwelling-house; and if deemed unfit ii) the reasons for unfitness.

Problem solving method: The associative method was chosen because this task can be viewed as a selection problem and the required knowledge can be formulated in terms of rules.

Sub-task proposal knowledge: Apply the task when the all fitness requirements have been assessed.

Search control knowledge: The associative method evaluates each statutory requirement and finds out if the property is or is not unfit.

Figure 4.6 shows part of the task structure for the assessment of the fitness of the property.

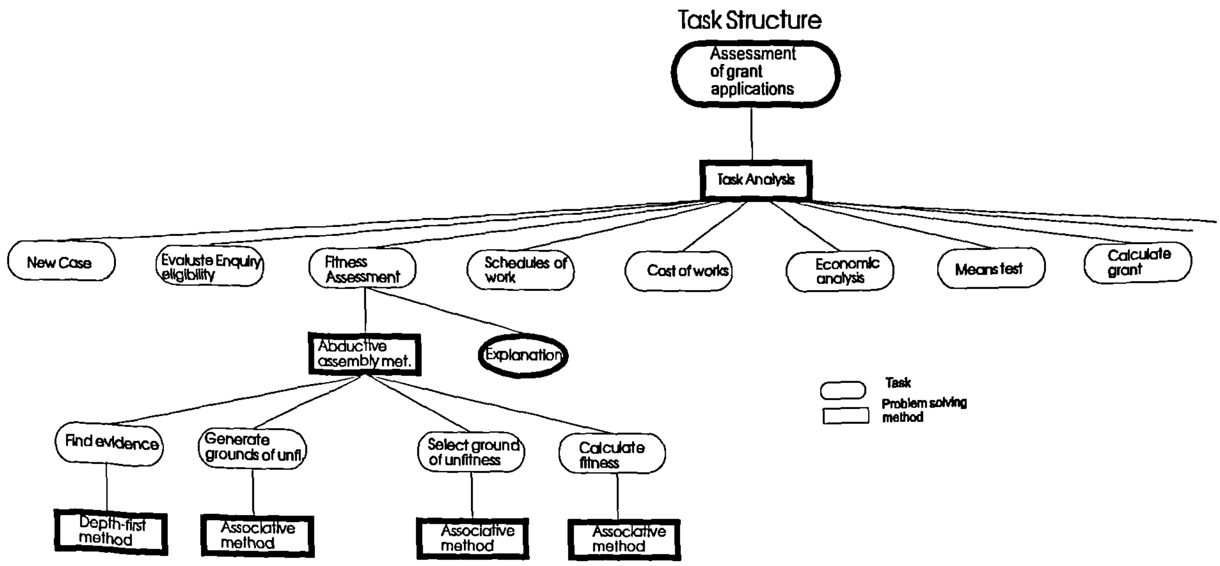


Figure 4.6: Part of task structure for the assessment of the fitness of the property

4.3.2.4.2- To Determine The Schedules Of Work

Definition: The task of determining the schedules of work task is specified by: i) the grounds of unfitness found during the assessment of the fitness of the dwelling-house; ii) a set of requirements from the fitness standard for human habitation; and iii) a technology of the available renovation solutions.

Initial state: The initial state includes information (corresponding to the goal state of the assessment of the fitness task) about: i) the age of the dwelling-house; ii) the fitness of the property; iii) the reasons for unfitness; and iv) the grounds of unfitness.

Goal state: The goal state includes a specification of the schedules of work required to repair the dwelling-house up to the required standard of fitness for human habitation.

Problem solving method: Past application cases can provide a basis for deriving the schedules of work required to make fit a given unfit dwelling-house. They hold a number of schedules of work which have been used in previous situations. Client's experts use these past schedules of work to solve new problems in similar situations.

Taking into account that: i) the main source of knowledge for the task of determining the schedules of work comes from past application cases; and ii) schedules of work which have been successfully used in the past can be applied to solve new problems in similar situations; the CBR method was selected to accomplish the task of determining the schedules of work. Therefore, in such situations, CBR is more computationally efficient than other possible methods such as the associative method, because it can derive a new schedule of work from an old one.

Sub-tasks: The CBR method decomposes the schedules of work task into three sub-tasks: i) *to select the most similar application case*; ii) *to retrieve a similar schedule of work*; *and if it fails to retrieve a similar application case*: and then iii) *to adapt schedules of work.*

Sub-task proposal knowledge: The task *to select most similar application case* is applied when the reasons for unfitness and its grounds are known. The task *to retrieve the similar schedule of work* is applied when a similar past application case is found. The task *to adapt a schedule of work to fit the specifications* is applied if a similar past application case is not found.

Search control knowledge: CBR accomplishes the schedules of work task by:

- selecting a most similar case from a set of candidate past application cases;
- retrieving the schedules of work that satisfy the specifications of the new application case; and
- if the method fails to select a similar case, then backtracking and adapting a schedule of work with solutions provided by the useful past applications cases.

A description of the each sub-task generated by CBR is given below.

4.3.2.4.2.1- To Select Similar Application Case

Initial state: The initial state includes information about: i) the age of the property; ii) the reasons for unfitness; and iii) the grounds of unfitness.

Goal state: The goal state is a similar application case.

Problem solving method: The retrieval of a similar application case can be seen as a selection problem where the specification of the current application case is matched it with the specifications of the stored application cases. Therefore, knowledge needed for the search and matching can be formulated in terms of rules. Thus, the associative method was selected to accomplish this task.

Search control knowledge: The associative method accomplishes the task by taking the specification of the current problem and matching with the specifications of a set of candidate application cases stored in the case library.

The selection of a similar application case is illustrated by figure 4.7.

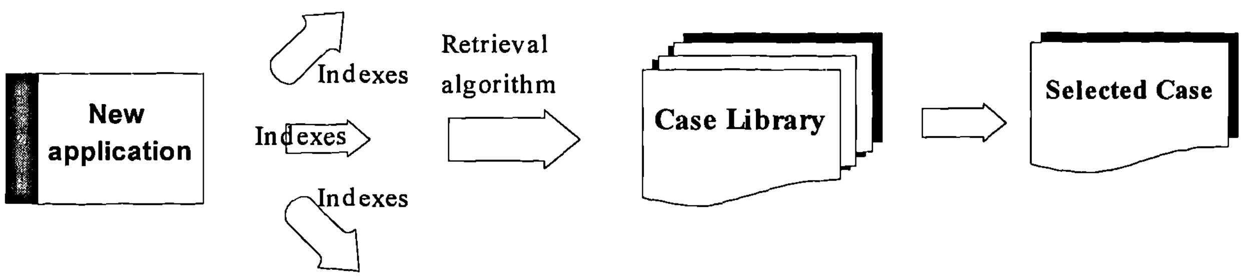


Figure 4.7: Selection of a similar application case

4.3.2.4.2.2- To Retrieve Similar Schedules Of Work

Initial state: The initial state includes a similar past application case.

Goal state: The goal state is a specification of the schedules of work.

Problem solving method: The retrieval of schedules of work can be seen as a selection problem where the specification of the current problem is matched with the specification of works stored in the similar case. Therefore, knowledge needed for matching can be formulated in terms of rules. Thus, the associative method was selected to accomplish this task.

Search control knowledge: The associative method accomplishes this task by taking the specifications of the failure and the defective building component of the current problem and matching them with the specification of works contained in the selected past application case.

4.3.2.4.2.3- To Adapt Schedules Of Work

Initial state: The initial state includes information about: i) the age of the property; ii) the reason for unfitness; iii) the ground of unfitness; and iv) a set of useful application cases.

Goal state: The goal state is a modified specification of the schedules of work.

Problem solving method: A specification of a renovation work can be decomposed into a number of different components. Thus, the adaptation of a renovation of work can be seen as a substitution problem. Substitution methods have been used in several CBR systems (CLAVIER, PERSUADER, JUDGE). Taking into account the specific nature of a schedule of work and the experience of other CBR systems, a substitution method was selected to accomplish the adaptation.

Search control knowledge: The substitution method performs the task by: Firstly, identifying the component of a schedule of work to be replaced. Secondly, taking the specification of the failure and defective building component and retrieving a set of useful schedules of work satisfying that failure and building component. Thirdly, suggesting to the user a set of similar types of work. Finally, implementing the adaptation.

Figure 4.8 shows part of the task structure for the task of determining the schedules of work.

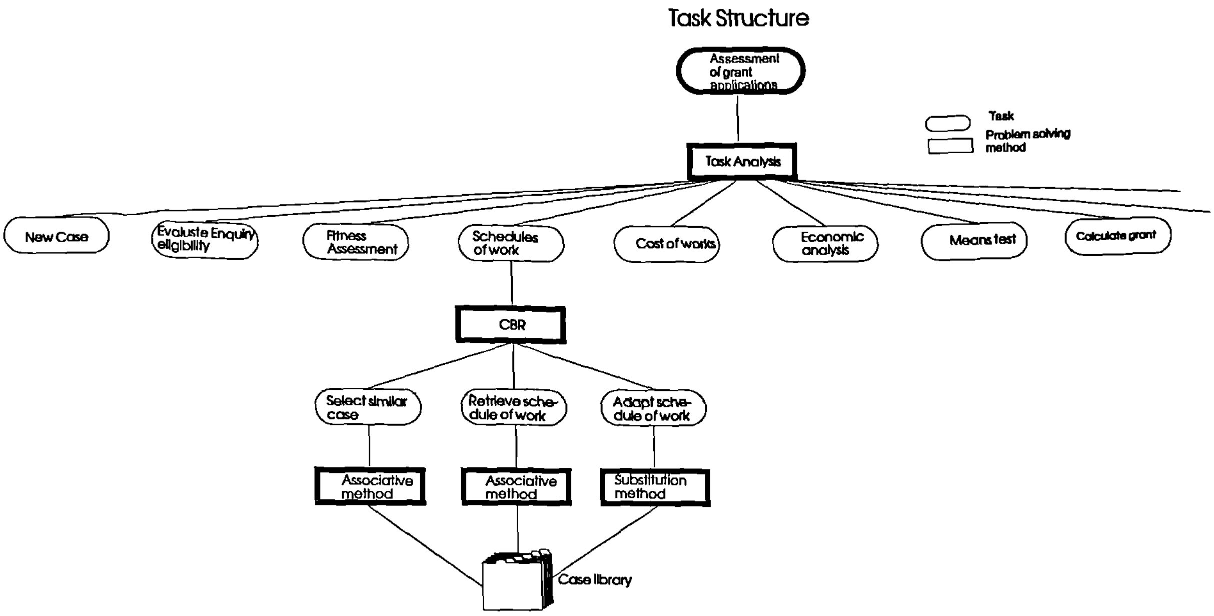


Figure 4.8: Part of the task structure for determining the schedules of work

4.3.2.4.3- To Undertake the Test of Resources -Means Test

Definition: The means test is specified by: i) the basic needs of the applicant and other occupants; and ii) the resources of the applicant and other occupants with an interest in the application.

Initial state: The initial state includes information about: i) the applicant and other occupants and ii) the financial resources of the relevant persons.

Goal state: The goal state includes information about the reduction in the amount of grant.

Problem solving method: The means test has different goals and tasks to be performed. The means test can be accomplished by decomposing it into sub-problems with much smaller problem spaces. Thus, the decomposition method was selected for the task.

Sub-tasks: The decomposition method decompose the means test task into four sub-tasks: i) *to determine relevant persons*; ii) *to determine applicable amount of the relevant persons*; iii) *to determine eligible income and capital of the relevant persons*; and iv) *to determine reduction of grant*. These sub-tasks are performed following a pre-specified procedure laid-down by the legal framework.

Search control knowledge: The decomposition perform the task by: i) generating specifications for the sub-problems in the decomposition: ii) solving each sub-problem; and iii) finding the reduction in the amount of grant.

The description of the sub-tasks mentioned above follows in the next sections.

4.3.2.4.3.1- To Determine the Relevant Persons

Initial state: The initial state includes the information about the composition, age and family relations of the applicant and other occupants.

Goal state: Then goal state includes the identification of the relevant persons to the current application to whom the mean test will apply.

Problem solving method: The associative method was chosen because this task can be viewed as a selection problem and the required knowledge can be formulated in terms of rules.

Sub-task proposal knowledge: Apply the task when data about the occupants is known.

Search control knowledge: The associative method accomplishes the task by matching the data about the occupants with the specification of the relevant person types stored in the system.

4.3.2.4.3.2- To Determine Applicable Amount of the Relevant Persons

Initial state: The initial state includes information about the relevant persons.

Goal state: The goal state includes the weekly applicable amount for each relevant person and the total weekly applicable amount related to all of them.

Problem solving method: The associative method was chosen because this task can be viewed as a selection problem and the required knowledge can be formulated in terms of rules.

Sub-task proposal knowledge: Apply the task when the relevant persons are known.

Search control knowledge: The associative method accomplishes the task by matching the data about the relevant persons with the types of applicable amounts stored in the system.

4.3.2.4.3.3- To Determine Eligible Income and Capital of the Relevant Persons

Initial state: The initial state includes information about financial resources of the relevant persons.

Goal state: The goal state is the one which includes the weekly eligible income and capital for each relevant person and the total weekly eligible income of all of them.

Problem solving method: The associative method was chosen because this task can be viewed as a selection problem and the required knowledge can be formulated in terms of rules.

Sub-task proposal knowledge: Apply the task when the financial resources of the relevant persons are known.

Search control knowledge: The associative method accomplishes the task by matching the data about the financial resources of relevant persons with the types of eligible incomes stored in the system.

4.3.2.4.3.4- To Determine The Reduction In The Amount Of Grant

Initial state: The initial state includes information about: i) the disposable income; ii) the previous contributions to grants; iii) type of certificate of future occupation which is provided.

Goal state: The goal state includes the amount by which the grant must be reduced.

Problem solving method: The associative method was chosen because this task can be viewed as a selection problem and the required knowledge can be formulated in terms of rules.

Sub-task proposal knowledge: Apply the task when the total applicable amount and total eligible income and capital are known.

Search control knowledge: The associative method accomplishes the reduction of the grant by: First, calculating the disposable income. Secondly, matching the disposable income and type of certificate with the types grant reductions stored in the system.

Figure 4.9 shows part of the task structure for the means test task.

4.3.2.4.4- To Find the Decision/Solution

Definition: The task of finding a decision for the current application case is specified by: i) a set of indices for retrieving a similar application case containing the decision; and ii) a set of requirements for presenting of the application case.

Initial state: The initial state includes information about: i) the specification of the desired decision/solution which includes information about: a) the degree of eligibility of the application; b) the area based planning issues where the property is situated; c) the relation of the property to adjoining properties; d) the neighbourhood characteristics; e) the means test; f) the level of fitness of the property; and ii) the requirements for presenting; and storing the application case.

Goal state: The goal state includes: i) the decision plan for the current application case; and ii) the current application case stored in the case library.

Problem-solving method: Past application cases can provide a basis for deriving the decision for the current application. They store decisions which have been used in previous situations. Taking into account that: i) the main source of knowledge for the task of finding the decision comes from past application cases; and ii) decisions which have been successfully used in the past can be applied to solve new problems in similar situations; the CBR method was selected to accomplish the task of finding the decision for the current application.

Sub-tasks: The CBR method decomposes the decision plan task into three sub- tasks: i) *to retrieve similar application case*; ii) *to present the current application case solved*; and iii) *to store the application case in the case library.*

Sub-task proposal knowledge: The task *to retrieve similar application case* is applied when the indices for the current task are known. The task *to present the application case solved* is applied when a similar past application case is found. The task *to store the application case in the case library* is applied after the solution plan has been presented.

Search control knowledge: CBR accomplishes the decision task by:

- selecting a most similar case from a set of candidate past application cases;
- presenting the current application case; and
- storing the current application case.

A description of the each sub-task generated by CBR is given below. Figure 4.9 shows part of the task structure for the task of finding the decision/solution.

4.3.2.4.4.1- To Retrieve Similar Application Case

Definition: Application case retrieval is specified by a set of indexes for guiding the search and matching it in the case library.

Initial state: The initial state includes the indices of the current application case.

Goal state: The goal state includes a similar application case.

Problem-solving method: The retrieval of a similar application case can be seen as a selection problem where the specification of the current application case is matched with the specifications of the stored application cases. Therefore, knowledge needed for the search and matching can be formulated in terms of rules. Thus, the associative method was selected to accomplish this task.

Search control knowledge: The associative method accomplishes application case retrieval by:

- searching and matching the most similar case in the case library; and
- retrieving the solution from the selected case.

4.3.2.4.4.2- To Present the Application Case

Definition: The application case once solved should be presented so as to maximise the usefulness of its information to the user. The presentation task is specified by: i) the appropriate information to display the solution plan; and ii) an appropriate format to present the information.

Initial state: The initial state includes the current application case.

Goal state: The goal state includes the desired information presented according a specific format.

Problem-solving method: An algorithm was chosen for the task.

4.3.2.4.4.3- To Store the Application Case

Definition: The application case should be adequately stored, in order to make it useful for future applications. Adding a new successfully processed application, is an inherent part of the system's learning process. The application cases storage task is specified by the structure of the case library.

Initial state: The initial state includes the current application case.

Goal state: The goal state includes the application case stored into the case library.

Problem-solving method: An algorithm was chosen for the task.

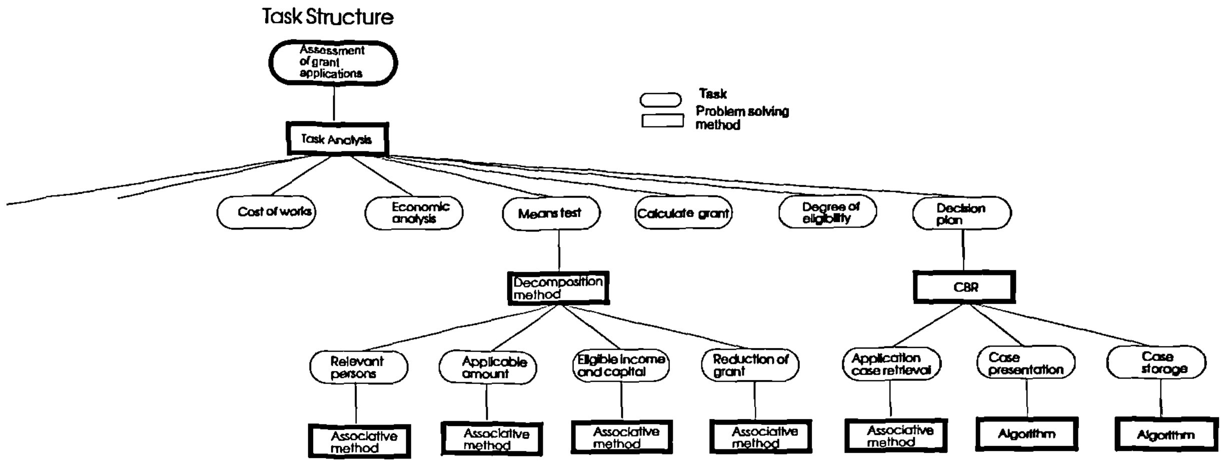


Figure 4.9: Part of the task structure for the test of resources and decision tasks

4.4- SUMMARY AND CONCLUSIONS

This chapter has outlined how the task structure analysis of the assessment of applications for the HRGS was carried out. The three versions of the task structure developed in this research are not complete. However, they can be easily reused in other domain problems such as: i) the assessment of applications for common parts grants; and ii) the assessment of applications for houses in multiple occupation grants. Tables 4.2, 4.3 and 4.4 present a summary of these versions.

The task structure produced as a consequence of the task analysis identifies: i) the tasks which CBR is responsible for; ii) other methods for the remaining tasks; and iii) which tasks application cases can be usefully applied to the problem solving.

At the lowest level of the task structure one method is used to perform a sub-task corresponding to another method. For instance, the associative method is applicable both to the retrieval of a similar application case and a solution plan in the CBR method. Similarly, the substitution method is used to perform the adaptation sub-task in the CBR method. According to Goel (Goel A., 1989), this use of one method for performing a sub-task in another method enables a *task-directed integration of methods*. Therefore, this provides a basis for designing the system's control of the processing.

The task structure is an analytical tool which provides a vocabulary to use: i) in specifying: what the system should do in terms of its goals and how the system accomplishes its goals; and ii) in guiding the acquisition of the knowledge needed to solve the different tasks of the assessment of applications for the HRGS.

As a consequence of the discussion presented in the above sections, a number of conclusions about the task structure were formulated. These conclusions are:

- The task structure describes the proposed system in terms of its knowledge content without being tied to the details of its implementation.
- The task analysis allowed the author to identify the reasoning approaches suited to meet the various needs of the assessment of applications for the HRGS. Therefore, it has shown that CBR alone is not ideally suited for all tasks of the assessment of grant applications. Other methods are required in combination with CBR to implement the proposed system.
- The task structure shows how different kinds of problem solving methods can be associated into a single framework for accomplishment of the overall task.
- Problem spaces in describing problem solving methods put together: i) tasks; ii) knowledge to perform the tasks; and iii) inferences which occur on that knowledge. This advantage of problem spaces was emphasised by Newell and Chandrasekaran (Newell A. and Chandrasekaran B., 1993).
- The task structure provides a basis for an architecture for the proposed system.
- Part of the system's knowledge which has been described by the task structure includes: i) search and control knowledge; and ii) knowledge to propose what task to perform next. Knowledge to perform the assessment of grant application tasks will be acquired and guided by the task structure.

- The task structure analysis decomposed the overall task into more manageable chunks of sub-tasks, to the level where they can be solved with knowledge easily available in the domain.
- By chunking knowledge into problem spaces, only the information required to achieve a goal is brought up. This feature provides a high degree of modularity of knowledge which will in turn facilitate the system's validation and maintenance.

The knowledge needed to solve the tasks identified by the task structure analysis must be acquired. In the next chapter the acquisition of this knowledge is outlined and discussed.

Table 4.2: Summary of the task structure for the assessment of renovation grant applications

	Tasks	***Method***	**Sub-tasks**	***Method***	**Domain model**
Assessment of applications for renovation grants	**New Appli. Generation**	*Associative*	Ask data		Case library
			Select format	*Associative*	
			Create applic.case	*Associative*	
	Enquiry Eligibility	*Associative*	Evaluate	*Associative*	Tree of conditions
	Fitness Assessment	*Abductive assembly*	Find eviden-ce of failur.	*Depth-first*	Fitness std.
			Gen. hypot.- of unfitness	*Associative*	
			Select grou. of unfitness	*Associative*	
			Calculate fitness	*Associative*	
	Schedules of work	*CBR*	Select simi. applic. case	*Associative*	Case library
			Retrieve sch. works	*Associative*	
			Adapt sch. of works	*Associative*	Building elements
	Cost of wor.	*Algorithm*			
	Economic analysis	*Associative*	Calc. NPVs	*Algorithm*	
			Econ.merit	*Associative*	
	Test of Resources-Means test	*Decomposition*	Relevant persons	*Associative*	
			Applicable amount	*Associative*	
			Eligible income	*Associative*	
			Reduction of grant	*Associative*	
	Calculate am. of grant	*Algorithm*	Calculate am. of grant	*Algorithm*	
	Degree of eligibility	*Associative*	Degree of eligibility	*Associative*	
	Decision plan	*CBR*	Retrieve simi. case	*Associative*	Case library
			Case presentation	*Algorithm*	
			Case storage	*Algorithm*	

Table 4.3: Summary of the task structure for the assessment of disabled facilities grant applications

Task	Sub-tasks	Method	Sub-tasks	Method	Domain model
Assessment of applications for disabled facilities grant	New Appli. Generation	*Associative*	Ask data		Case library
			Select format	*Associative*	
			Create applic.case	*Associative*	
	Enquiry eligibility	*Associative*	Evaluate	*Associative*	Tree of conditions
	Assess. of adaptation facilities	*Abductive assembly*	Necessary & appropriate	*Depth-first*	
			Reasonable & practica.	*Depth-first*	
	Schedules of works	*CBR*	Select simi. applic. case	*Associative*	Case library
			Retrieve sch. works	*Associative*	
			Adapt sch. of works	*Associative*	Building elements
	Cost of wor.	*Algorithm*			
	Test of Resources-Means test	*Decomposition*	Relevant persons	*Associative*	
			Applicable amount	*Associative*	
			Eligible income	*Associative*	
			Reduction of grant	*Associative*	
	Calculate am. of grant	*Algorithm*			
	Degree of eligibility	*Associative*			
	Decision plan	*CBR*	Select simi. app. case	*Associative*	Case library
			Case presentation	*Algorithm*	
			Case storage	*Algorithm*	

Table 4.4: Summary of the task structure for the assessment of minor works assistance applications.

Task	Sub-tasks	*Method*	Sub-tasks	*Method*	Domain model
Assessment of applications for minor works assistance	**New Appli. Generation**	*Associative*	Ask data		Case library
			Select format	*Associative*	
			Create applic.case	*Associative*	
	Enquiry eligibility	*Associative*	Evaluate		Tree of conditions
	Schedules of works	*CBR*	Select simi. applic. case	*Associative*	Case library
			Retrieve sch. works	*Associative*	
			Adapt sch. of works	*Associative*	Building elements
	Cost of wor.	*Algorithm*			
	Calculate am. of grant	*Algorithm*			
	Degree of eligibility	*Associative*			
	Decision plan	Select simi. app. case	*Associative*		Case library
		Case presentation	*Algorithm*		
		Case Storage	*Algorithm*		

CHAPTER 5

KNOWLEDGE ACQUISITION

5.1- INTRODUCTION

The task structure outlined in chapter 4 decomposed the overall task for the assessment of applications for the HRGS into several sub-tasks. Each one of these sub-tasks is performed by searching through a problem space from a path from the initial state to the goal state. This search requires knowledge in order to map the initial state to the goal state of the task being solved. It consists of the all knowledge needed to solve the task. Acquiring and organising this knowledge is an essential step in completing the task structure for the assessment of grant applications outlined in chapter 4.

In this chapter, the discussion is about how the task knowledge is acquired in such a way that can be effectively represented in the system proposed in this research. Thus, this chapter begins with a discussion of the main knowledge acquisition methods, followed by a description of the available sources where the task knowledge can be acquired. This chapter finishes with a general description of the knowledge required to perform the tasks described by the task structure.

5.2- KNOWLEDGE ACQUISITION METHODS

5.2.1- BACKGROUND

Some of the literature explicitly addresses the topic of knowledge acquisition (KA) as a phase within the KBSs development life cycle, consisting of several stages. Forsythe and Buchanan (Forsythe D. and Buchanan B., 1993), defined KA as a process involving: i) gathering information from one or more experts and from documentary sources; ii) ordering that information in some way; and iii) translating it into machine-readable form. Another definition for KA is given by Schreiber, Wielinga and Breuker (Schreiber G., et al., 1993). According to these authors, KA involves at least the following activities: i) eliciting the knowledge in an informal verbal-form; ii) interpreting the elicited data using some conceptual framework; and iii) formalising the conceptualisation in such a way that the program can use that knowledge. A more recent view of KA is the model from Birmingham and Klinker (Birmingham W. and Klinker G., 1993). According to these authors KA, within the life cycle of KBSs, has the goal of eliciting, organising and populating a knowledge base with information required by the selected problem-solving approaches.

In this chapter, KA is assumed to be the process of eliciting and organising the task knowledge following the task structure outlined in chapter 4 and, it is viewed as part of a wider modelling activity which is independent of the implementation language.

5.2.2- SOME KA METHODS

KA has attracted a lot of research towards the development of methods to assist KBS designers. Motoda (Motoda H., 1994) pointed out that the first generation of KBSs mostly used knowledge elicitation techniques for KA regarded as rule or frame acquisition. With the advent of knowledge modelling level approaches, KA can no longer to be regarded as purely rule or frame acquisition. Recently, a number of KA methods and tools supported by task-based models have been developed. These methods provide techniques to elicit, acquire and organise knowledge for a variety of real-world tasks. According to Yost (Yost G., 1993), the task-based KA methods are becoming increasingly popular for the development of KBSs.

Table 5.1 summarises some of the most important KA methods classified by the presupposed model, elicitation procedures and implementation tool. A detailed description of these methods is presented in Appendix 6.

Table 5.1: Some major KA methods

Method	Presupposed model	Elicitation procedure	Implementation	Description
1.Knowledge roles	task structure based	elicit repetitive process of problem-solving domain	manual/computer	uses the knowledge roles, determined by the problem-solving methods to guide the KA
2.*Task specific*	task structure based	elicit repetitive process of problem-solving domain	manual/computer	uses the information about the structure of a task-specific structure to guide the KA
3.*KADS*	multilevel knowledge modelling and representation	structured interview; protocol analysis	manual/computer	the description of the various models guide the KA

Method	Presupposed model	Elicitation procedure	Implementation	Description
4.***Interviewing***	general	structured interview	manual	ask general questions and record as much as possible
		focused interview	manual	interview with open questions and a list of topics to cover
		unstructured interview	manual	interview with strict agenda and list of specific questions
5.***Implicit knowledge structures***	implicit knowledge structures	recall expert problem-solving behaviour in domain	manual/computer	record the expert behaviour in solving a problem or performing a task
6.***Entity-attribute grids***	knowledge characterised by distinctions made	elicit entities in domain and distinctions between them	manual/computer	use distinctions of knowledge to guide the KA
7.***Multiple experts***	general	delphi	manual/computer	gather information from people independently
		multiple sources	manual/computer	gather information from multiple sources separately and combine for use
8.***Causal model***	causal model	interviewing; repertory grid; and protocol analysis	manual	uses causal models to guide KA
9.***Learning-based techniques***	analogy;	interview, observations	manual/computer	apply knowledge from old situations in similar new situations
	induction of models from experience;	interview, observations	manual/computer	generate models from the past experience
	similarity-based learning	interview, observations	manual/computer	learn similarities and differences from sets of examples
10.***Text knowledge acquisition and analysis***	natural-language understanding	natural language understanding; chart parser; syntactic analysis; text grammars; domain-independent meta-language	manual/computer	knowledge directly from text
11.***Case acquisition methods***	case-based model	case-based parsers; case-based language understanding	manual/computer	acquire knowledge from past cases

5.3- FRAMEWORK FOLLOWED FOR KA

5.3.1- KNOWLEDGE SOURCES IN THE SYSTEM'S DOMAIN

The study of the Client's procedures, revealed that much of the knowledge required to perform each sub-task of the assessment of applications for the HRGS comes from a number of sources such as: regulatory and technical texts, expert's knowledge, computer databases and past application files. Figure 5.1 shows the main sources of knowledge for the assessment of applications for the HRGS. These sources are detailed below as follows:

From texts:

- *Statute and regulatory texts*- Acts of Parliament and statutory instruments; circulars, guides and booklets from Government bodies; guides, procedures and planning provisions from L A and other agencies.
- *Technical and normative texts*- technical publications from specialised agencies and professional organisations -like BRE, CIRIA, PSA, BS, NBS, RICS; technical publications from experts and researchers in the field.
- *Files containing records of past applications*- LAs have stored a number of files of individual applications. These files contain detailed information of past applications organised in accordance with the procedure followed during the assessment. The information they contain underlies the problem-solving strategies used by the human expert during the decision-making process.

From computer databases:

- *Data on individual applications*- The computer databases store data on individual applications in the processing stage.

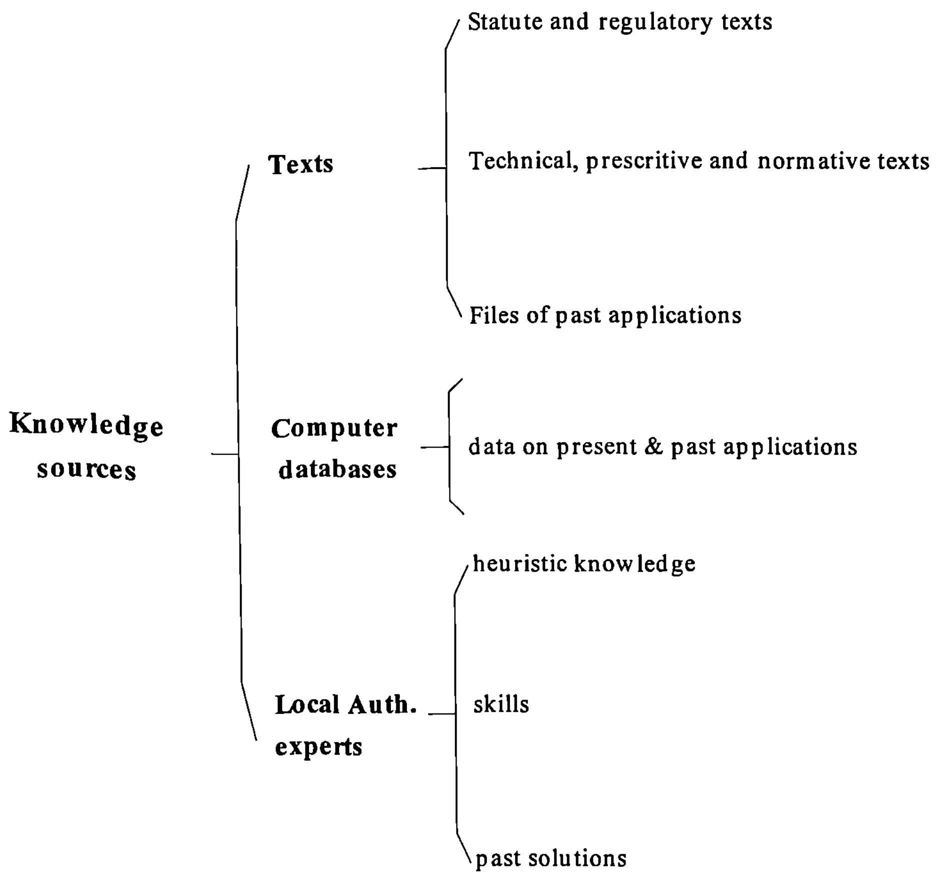


Figure 5.1: Main sources of knowledge for the assessment of applications for the HRGS

From human experts:

- *Domain specific heuristic knowledge*- Rules of thumb.
- *Skills*- Client's experts can display knowledge about how to perform the task.
- *Past solutions*- Experienced experts can remember past solutions when confronted with practical situations.

5.3.2- THE FRAMEWORK FOR KA

The KA followed a framework which includes five components: i) the task to be performed; ii) the initial and goal states of the task; iii) the available source of knowledge; iv) the available knowledge acquisition method; and v) the system's implementation stage. Figure 5.2 shows the relationship between these components in the framework.

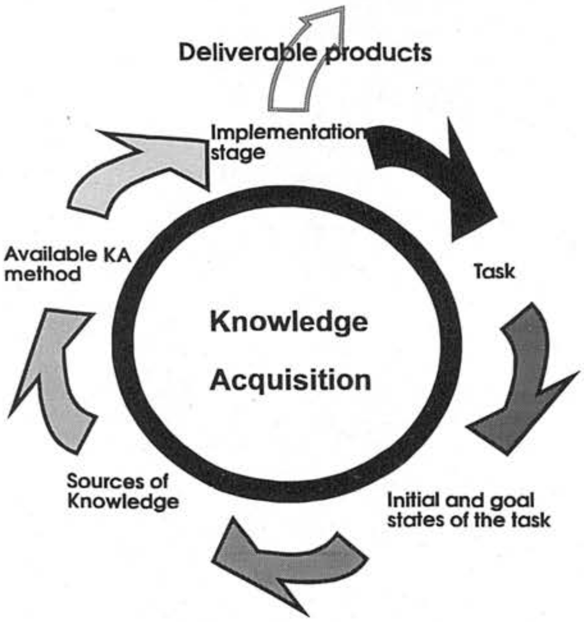


Figure 5.2: The five components of the KA framework

Table 5.2 lists the KA methods used at each stage of the system's implementation cycle.

5.4- ACQUISITION OF PAST APPLICATION CASES

How the application cases can be acquired and modelled in the case library of the system was a key issue in all KA processes followed in this research. The number of useful past applications available in the domain is large.

Kriegasman and Barletta (Kriegsman M. and Barletta R., 1993), stressed that a robust case library, containing a representative and well distributed set of historical cases, is the foundation for a good system.

Table 5.2: KA methods employed at each system's development cycle

KA method	**Development Methodology - Client Centred Approach**			
Postal survey; and interviewing	***Holistic picture***			
Task-based method; text acquisition; and case-acquisition		***Skeleton system***		
Task-based method; text acquisition; and case-acquisition			***Demo system***	
Interviewing; and case-acquisition				***Working system***

Past applications form a main source of knowledge. They can provide 'ballpark' solutions to new similar situations, avoiding the time necessary to derive those solutions from scratch. Kolodner (Kolodner J., 1993) pointed out that "accumulating the right cases, extracting the appropriate knowledge from them, and making sure cases are indexed advantageously results in several different enhanced performance behaviours:

- the ability to perform in more situations;
- increased efficiency in familiar situations;
- increased ability to cope in problematic situations; and
- increased ability to take advantage of opportunities as they arise".

The Client has stored hundreds of past applications that have been successfully processed since the HRGS came into force. The acquisition of knowledge stored in past applications in terms of cases was carried out in accordance with the following basic phases based on the Kitano (Kitano et al., 1993) case acquisition method:

1.Collection of seed application cases into full-text case reports:

The first step consisted of collecting a number of seed cases from a sample of over 300 past applications. Thus, the goal of this phase was to collect a set of past applications which could be modelled as seed cases covering applications for: i) renovation grants; ii) disabled facilities grants; and iii) minor works assistance.

This phase consisted of four main steps:

- *Collection of a representative sample of past applications-* this sample of past applications was used: i) to define the case categories and case report formats; and ii) to provide the seed application cases.
- *Definition of case categories-* from this sample the categories of applications processed by the Client were identified.
- *Definition of case report formats-* case report formats were defined for each case category.
- *Selection of seed application cases-* 60 most representative past applications were selected from a sample of 300 past applications.
- Finally, *full-text case reports* were created for the 60 seed application cases.

The seed application cases were chosen by considering the need to cover as broad a range of application contexts as possible. In so doing, the system should be able to reason in a broader range of new applications at an early development stage. Figure 5.3 shows the procedure followed for this phase of case acquisition. A sample of a case report form is presented in Appendix 7.

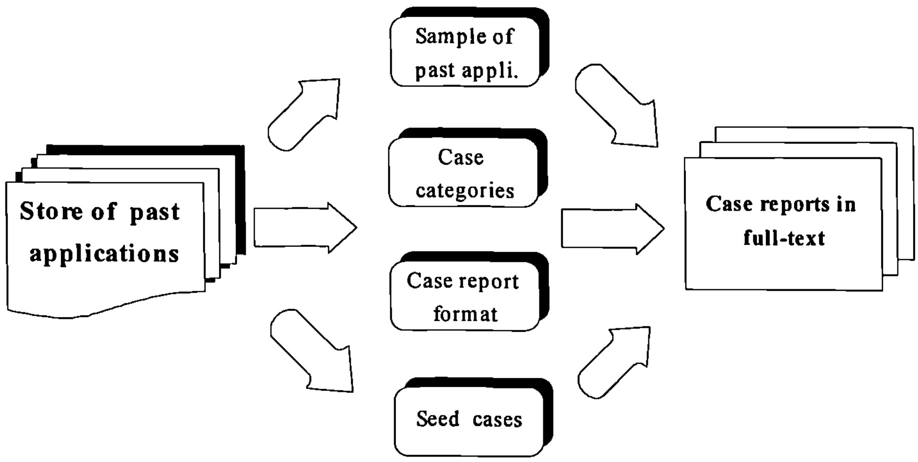


Figure 5.3: Procedure followed to create case reports

2.Identification of case representation formats:

This phase was concerned with establishing a case structure format for each case category. The goal of this phase was to extract all possible features from the case reports which could be used for the case representation and indexing. This phase was carried out in the following three steps:

- *Identification of case features (f_1, f_2,...f_i,..f_n)*- each feature was selected from the case reports after being examined to establish whether or not it is independent from other features.
- *Value grouping*- value(s) stored in the case reports was(were) assigned to the selected features of each case according to its function in the case library.
- *Indexes identification*- This step was concerned with selecting an indexing scheme that allows the system to recognise that a past application case is applicable to the current situation description. The goal of this step is to find, within the selected feature, the indexes that are most useful for: a) organising the application cases into the case library; and b) allowing an efficient search and matching of a similar application case in the case library.

This phase can be illustrated by figure 5.4.

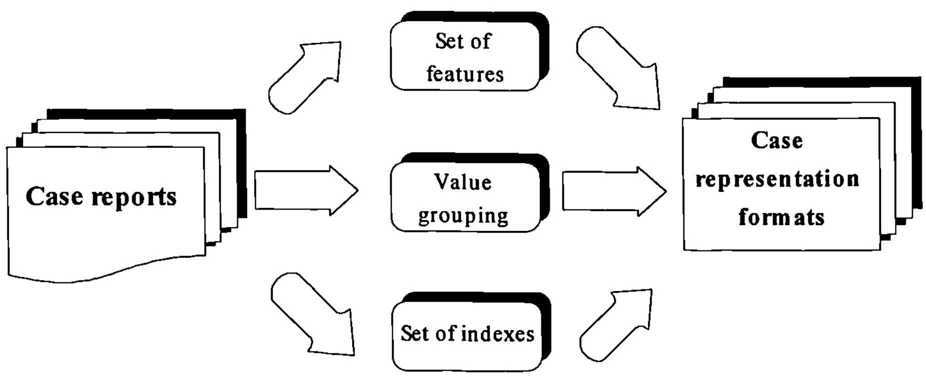


Figure 5.4: Procedure followed for case formats elaboration

3.Identification of the matching procedures:

This phase had the goal of finding: i) which features of an application case are important to focus on for judging similarity for retrieval purposes; and ii) what type of matching procedures will be used for comparing the new situation with those described in the past application cases. Figure 5.5 shows this phase, including its steps.

4.Testing and refining the case representation formats and indexing scheme:

The case representation formats needed to be tested and refined. The usefulness and accuracy of both case representation and case indexing during the system's validation were tested as it is described in chapter 8.

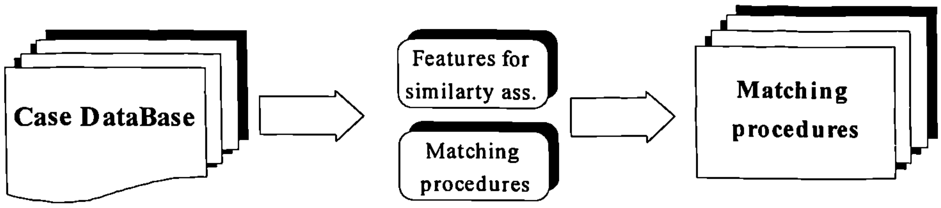


Figure 5.5: Procedure for extracting the matching procedures

Since CBR is a method that combines reasoning with learning (Kolodner J., 1993), the case acquisition process proceeded with the system development cycle. With more acquired application cases, the efficiency and competence of the system has increased. The case acquisition process described in this section is detailed later in chapter 7 where the implementation of the system is discussed.

5.5- GENERAL DESCRIPTION OF KNOWLEDGE

This section presents a description of some of the most important pieces of the knowledge required for performing the sub-tasks described in the task structure. This description consists of the task knowledge.

5.5.1- KNOWLEDGE FOR EVALUATING THE ENQUIRY ELIGIBILITY

The factors governing the enquiry eligibility differ with each type of grant. The main factors governing the eligibility of an enquiry for a renovation grant are:

1. *The age of the property*- grant applications other than for disabled facilities grant may not, however, be entertained in respect of properties built or provided by conversion less than ten years before the date of the current application.

2. *The interest-* any person with an owner's interest in all the "land" on which it is proposed to carry out works may apply for a grant.
3. *Type of tenure of the property-* define who can apply for renovation grants. Figure 5.6 shows who can apply for each type of grant.

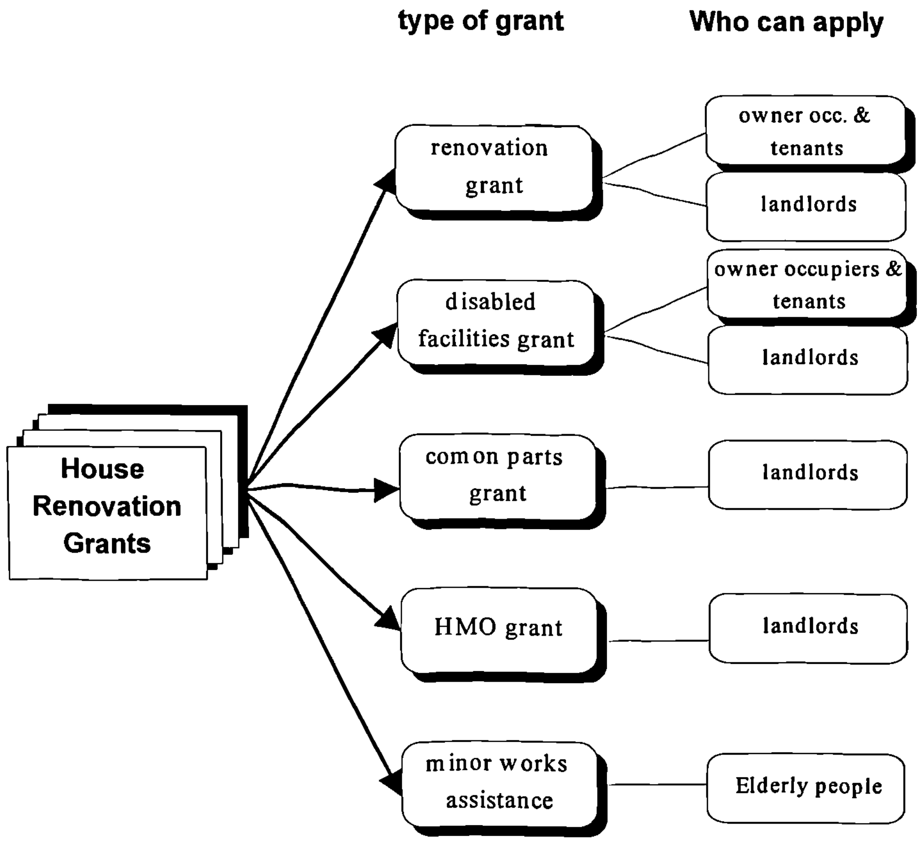


Figure 5.6: Who can apply for each type of grant

4. *State of works-* An enquiry for a renovation grant may not proceed if the works are completed before the application has been approved, unless the works are necessary to comply with a statutory notice. Where the works commenced but are not completed before the application is determined, other than where these are required to comply with a statutory notice, the application should be refused except where it is found that there were good reasons for beginning the works before approval.
5. *Nature of works-* Enquiries are only eligible where the nature of works are required to make fit the property of that enquiry.
6. *Residence-* The applicant must be wholly or mainly resident in the property.

The main factors governing the eligibility of an enquiry for disabled facilities grants are:

1. *Status of the disabled person for whom the grant is sought-* Only persons who are registered as a disabled person are eligible for disabled facilities grants.
2. *The interest* - As described for renovation grants.
3. *Type of tenure of the property-* As described for renovation grants.
4. *Nature of adaptation works required-* The disabled facilities grant is mandatory for works needed to make a property suitable for the disabled person's occupation.
5. *Residence-* As described for renovation grants.

The main factors governing the eligibility of an enquiry for minor works assistance are:

1. *Benefits in receipt by the applicant-* Either the applicant, or his partner, must be in receipt of income support, family credit, housing benefit or council tax benefit, in order to be eligible for minor works assistance.
2. *The interest-* The same as for renovation grants.
3. *Purposes of required works-* The minor works assistance shall only be given for small scale repair and or improvement works.
4. *Residence-* As described for renovation grants.

5.5.2- KNOWLEDGE FOR ASSESSING THE FITNESS OF THE HOUSE

As mentioned in chapter 2 a dwelling-house is fit for human habitation unless, in the opinion of the LA, it fails to meet one or more of the requirements set out by the fitness standard introduced by the Local Government And Housing Act 1989 (Act 1989) and, by reason of that failure, is not suitable for human occupation.

In deciding whether or not a property is unfit, a LA should determine for each of the statutory requirements whether or not the property is reasonably suitable for occupation. In reaching a decision about the fitness, LAs are asked to have regard to the guidance notes set out by the legal framework. For each of the requirements contained in the standard of fitness, the guidance provides general advice and lists the main items to which the LAs are asked to have regard in forming their opinion on the fitness of any property. It then provides advice on the determination of fitness in respect of some of the more typical defects which are found. The items to be regarded in forming an opinion on the fitness of a dwelling-house are presented in Appendix 8.

When the fitness standard is objectively interpreted and the assessment of the fitness of a property is properly performed may contribute for: i) ensuring an accurate assessment of the condition of the dwelling-house; ii) reducing or even avoiding the risk before any major capital commitment; iii) ensuring the successful accomplishment of renovation works; and iv) allowing a fair and equal access to mandatory renovation grants.

On the basis of the guidance notes set out by the legal framework, the Client has developed a work-document designed to assist its officers in assessing the fitness of any dwelling-house. This work-document lists a set of typical failures, presented as questions, to be regarded when inspecting a property and the corresponding decisions. Examples of these questions are presented below.

First example:

Requirement: It is structurally stable

1. Is the chimney stack unstable? Yes = Unfit
2. Is the roof badly sagged? Yes = Unfit
3. Are any ceilings liable to collapse? Yes = Unfit

Second example:

Requirement: It is free from serious disrepair

1. Are there slipped, tagged or broken slates? Yes = Marginal or Serious
2. Are any chimney pots dislodged or insecure? Yes = Serious
3. Are any outbuildings or yard walls badly open jointed or have part missing or perished brickwork? Yes = Marginal

This work-document provides instructions on how to reach a decision about the fitness of a building component with respect to each fitness requirement. The decision about the fitness of a building component can assume different values such as, unfit, serious, and marginal. Thus, a dwelling-house is unfit regarding any requirement if i) any single item is recorded unfit or serious in that requirement; and ii) 5 or more items are recorded as marginal, by their combined effect, the house will be deemed unfit in that particular requirement.

The following components are implicitly in each question of the Client's work-document: i) the fitness requirement; ii) the type of failure to be checked; iii) the building component; iv) the severity of failure; and v) the decision about the fitness. Examples of these components are shown by figures 5.7 and 5.8.

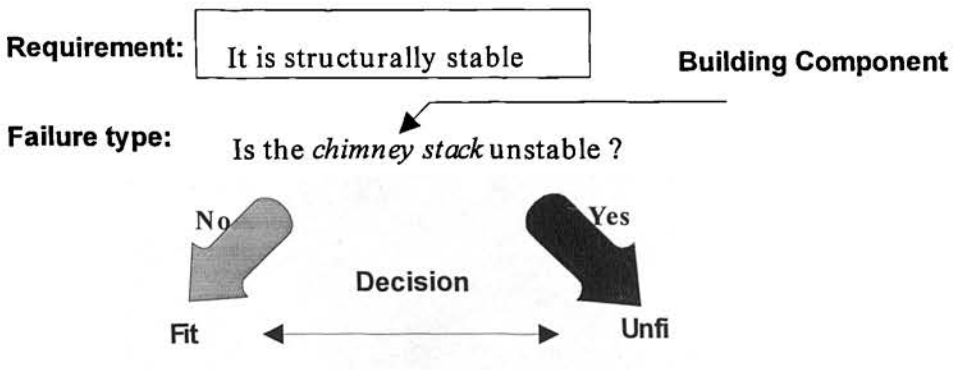


Figure 5.7: A question to find the evidence of failure of the chimney stack to the structural stability requirement

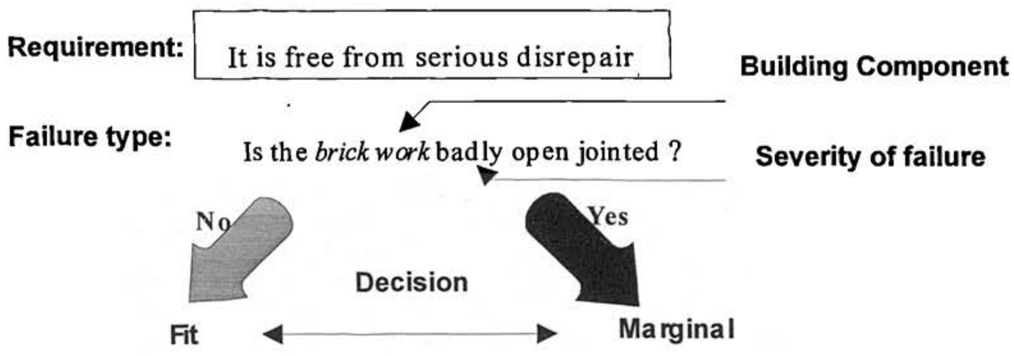


Figure 5.8: A question to find the evidence of failure of the brickwork to the repair requirement

An examination of 300 granted past applications for mandatory grants (dwellings deemed as unfit) has shown that the most common reasons for unfitness are a failure to comply with the requirements for: repair (97%), dampness (76%) and structural stability (55%). Table 5.3 shows the distribution of the unfit properties by the requirements which they failed to comply with.

Table 5.3: Distribution of past applications by reasons for unfitness in a sample of 300 unfit dwelling-houses

Reasons of unfitness (failed requirements)	**Percentage of properties %**
Stability	55
Repair	97
Dampness	76
Lighting	31
Heating	3
Ventilation	3
Cooking	28
Water Supply	7
Water Closet	10
Bath	10
Drainage	14

To assist in the assessment of the condition of any dwelling-house in searching for evidence of failures, a chart was developed in the context of this research. This chart maps building components and requirements of the fitness standard to a set of possible failures following a procedure shown in figure 5.9. This chart is partly illustrated by figure 5.10.

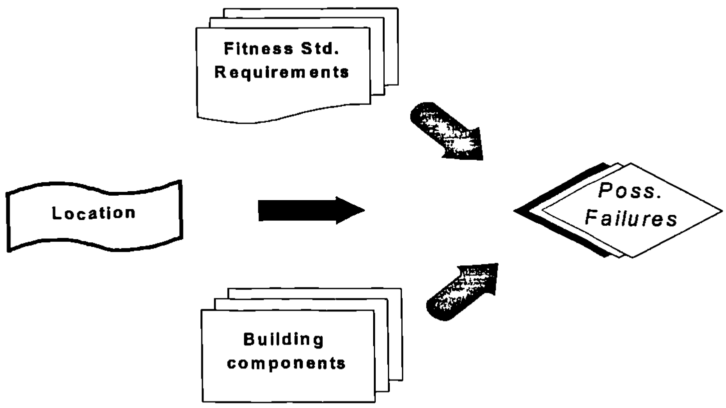


Figure 5.9: How fitness requirements and building components relate to failures

The experience shows that evidence of failure of the fitness standard is associated with one or more grounds of unfitness. The process of generating hypotheses of grounds of unfitness which will explain a given evidence of failure can be illustrated by figure 5.11.

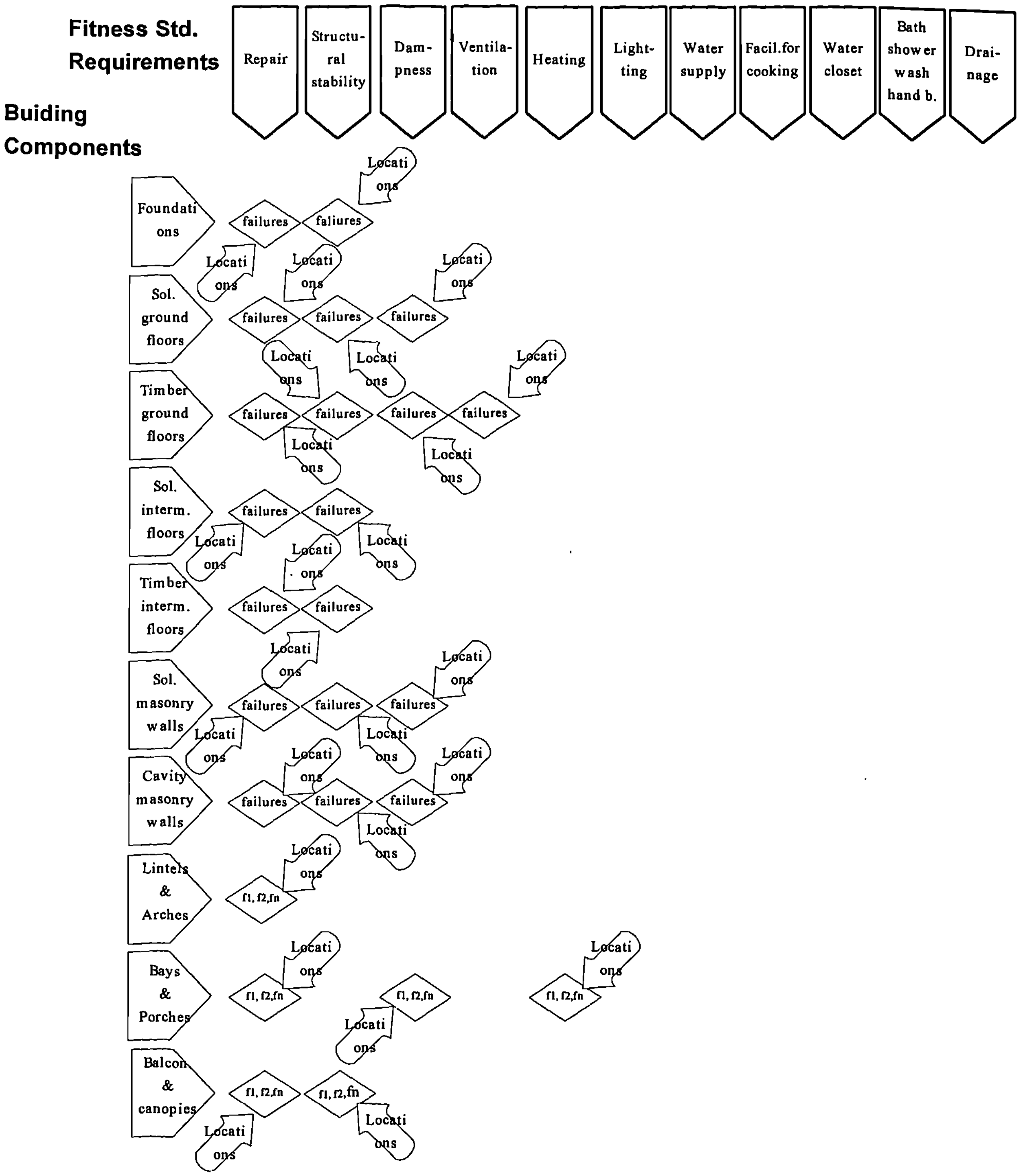


Figure 5.10: Chart developed to assist the search for evidences of failures

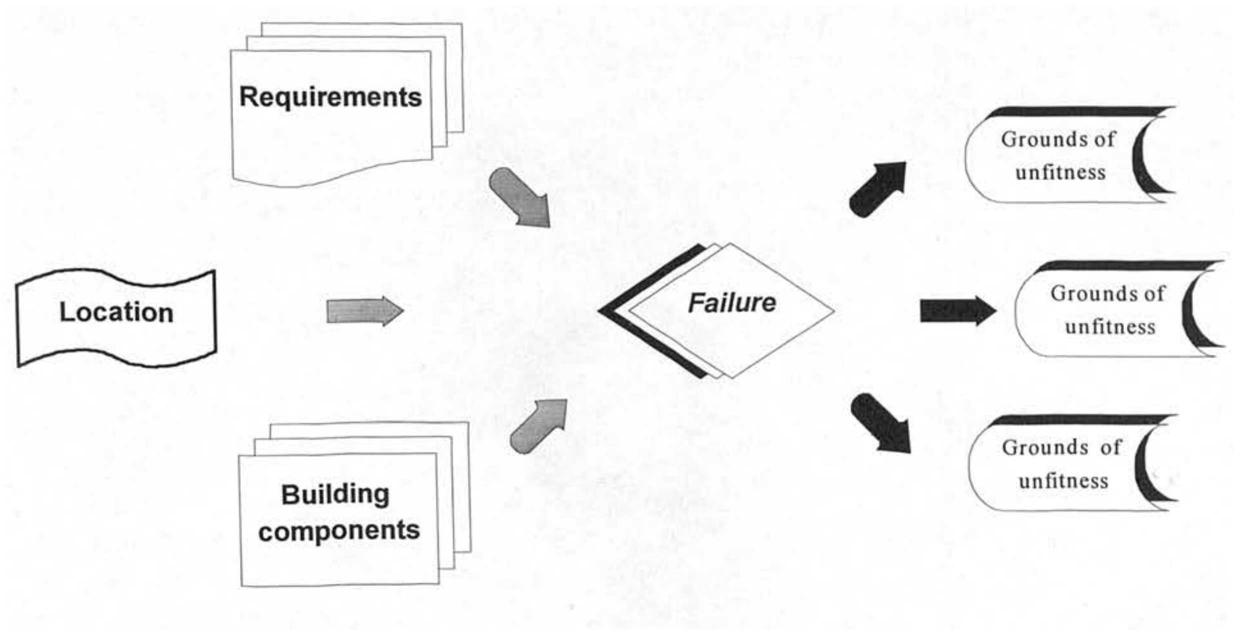


Figure 5.11: Process of generating hypotheses of grounds for unfitness.

An example of how grounds of unfitness are generated is shown in figure 5.12.

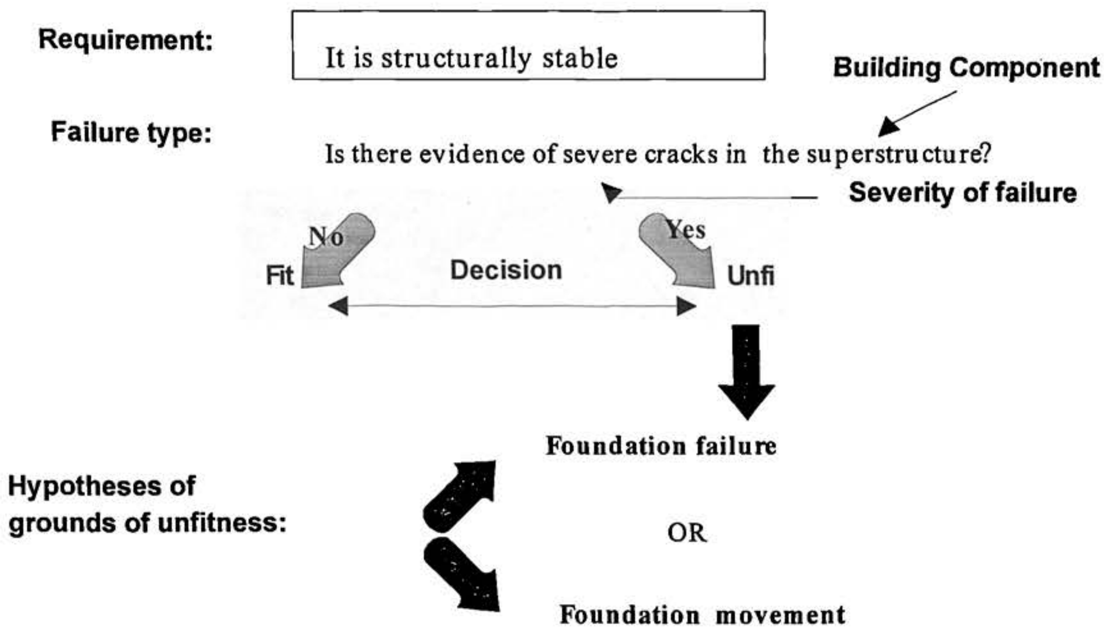


Figure 5.12: Hypotheses of ground of unfitness for the evidence of cracks in the superstructure

Past applications show that there are strong similarities between grounds of unfitness found in dwelling-houses which have been deemed unfit for the same reasons.

On the basis of the information stored in the sample of over 300 past applications, a table was built which relates evidences of failures and building components to the grounds of unfitness. An example of this work is presented in table 5.4.

Table 5.4: Partial view of the table that relates building components and evidence of failures to grounds of unfitness

Fitness Requirement	Building Component	Evidence Of Failure	Ground Of Unfitness
Structural Stability	**Foundations**	Evidence of severe cracks in the superstructure	1.Foundation failure. OR 2.Foundation movement.
	Solid masonry walls	Evidence of brickwork bulged	1.Brickwork over-restrained. OR 2.Brickwork severely fractured. OR 3.Brickwork badly deteriorated.
	Cavity masonry walls	Evidence of brickwork bulged	1.Wall leaves untied. OR 2. Walls over-restrained.
	Pitched roof	Evidence of roof badly sagged	1.Pitched roof timbers rotten. OR 2.Pitched roof timbers of inadequate size. OR 3.Pitched roof timbers over stressed.
Repair	**Solid ground floors**	Evidence of floor broken and uneven	1.Floor screeds broken and weakened. OR 2.Floor steel reinforcement corroded.
Dampness	**Solid ground floors**	Evidence of persistent dampness on ground floor	1.Solid ground floor persistent dampness due to absent DPM. OR 2.Solid ground floor persistent dampness due to faulty DPM. OR 3.Solid ground floor persistent dampness due to defective link between DPCs and DPM.
	Cavity Masonry walls	Evidence of penetrating dampness	1.Cavity masonry wall penetrating dampness due to defective pointing. 2.Cavity masonry wall penetrating dampness due to defective brickwork. 3.Cavity masonry wall penetrating dampness due to faulty DPCs.

Knowing the hypotheses of grounds of unfitness, judgement selects the most likely ground of unfitness that explains the current evidence of failure to the fitness standard.

Past applications have shown that a typical ground of unfitness generally includes four primitive components, as shown in figure 5.13. These components are: i) the building component; ii) the severity of failure; iii) the type of failure; and iv) the location of the failure.

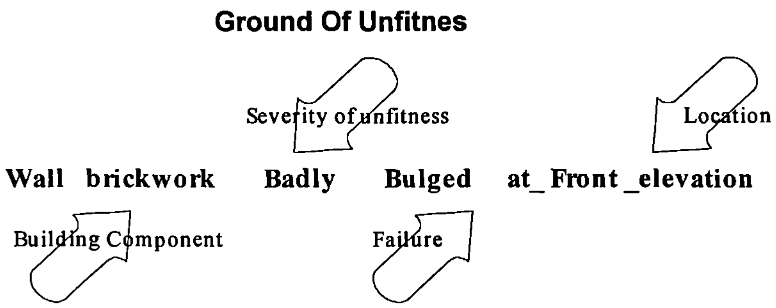


Figure 5.13: Example of a ground of unfitness in the structural stability requirement

Once the dwelling-house has been assessed under every requirement of the fitness standard, the overall fitness should be evaluated. When a property is deemed unfit then the reason(s) for its unfitness should be known. The reasons for unfitness may be: i) structurally unstable; ii) serious disrepair; iii) prejudicial dampness; iv) inadequate lighting; v) inadequate heating; vi) inadequate ventilation; vii) inadequate water supply; viii) unsatisfactory cooking facilities; ix) has no provision of a water closet; x) have no provisions of a bath or shower and wash hand basin; and xi) ineffective drainage.

5.5.3- KNOWLEDGE FOR DETERMINING THE SCHEDULES OF WORK

Past applications store a number of schedules of work which were successfully completed (according to the reports contained in the application files). These schedules of work were specified to remedy the grounds of unfitness which made the property unfit for human habitation.

When any property is deemed unfit for human habitation, a detailed specification of schedules of work to make it fit is required. The prime aim of this specification is to provide the contractors with an unambiguous definition of works to be carried out in a manner that is readily priceable and which can be used for the administration and supervision of the renovation action. A well-written and concise specification of schedules of work is a major concern of the Client's experts.

Past applications show that the schedules of work carried out and the costs of making fit an unfit dwelling-house were associated with a number of features. These features include: i) *the construction date of the property*; ii) *the type of property*; iii) *the type of tenure*; vi) *the reasons for unfitness*; v) *the severity of unfitness* (number of reasons for unfitness); and vi) *the grounds of unfitness*. Looking at the past applications, a strong similarity was found between the schedules of work of properties with same values for those features. These features can be used to guide the search and matching of a similar application case in the case library when the system is performing the task of determining the schedules of work.

Schedules of work found in past applications generally state the following primitive components:

- The type of work: a definition of the type of work required.
- The failure: an inclusion of the failure to be eliminated by the schedule of work.
- The severity of failure: the severity of degradation of the building component.
- The defective building component: a definition of the building component where the failure was identified.
- The location: an indication of the location of the failure.

And, when the information is available:

- The size: as far as possible an indication of all dimensions and quantities of work.

Examples of the structure of schedules of work found in past applications are given in figures 5.14 and 5.15.

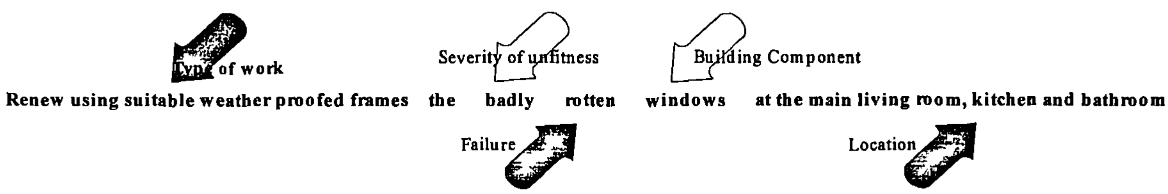


Figure 5.14 Structure of a schedule of work for repairing windows

Each component contributes to specifying the schedule of work.

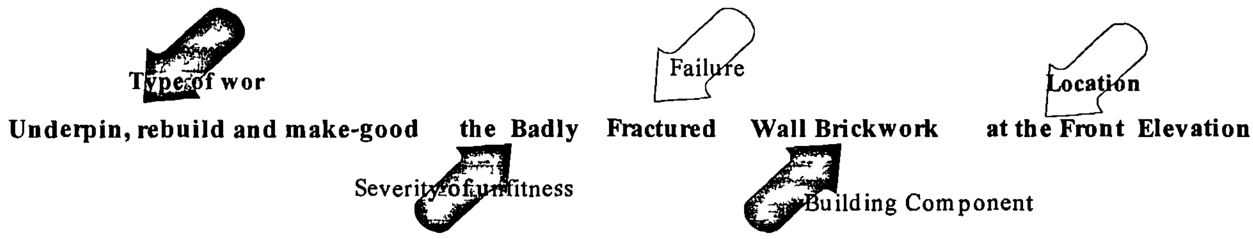


Figure 5.15: Structure of a schedule of work for repairing brickwork

Taking the example of repairing a badly fractured solid masonry wall, then it can be described in terms of its primitive components as follows:

- **Possible types of renovation works:** i) Take down, rebuild and make good; or ii) Underpin, rebuild and make good; or iii) Improve foundation support.
- **Severity of unfitness:** Badly.
- **Type of failure:** Fractured.
- **Building component:** Solid masonry wall (brickwork).
- **Location:** Continuous area of front elevation.
- **Size:** 20 m2.

Within the same ground of unfitness, by modifying the type of work component, different specifications of schedules of work can be obtained. Therefore, different specifications of works will produce different outputs such as, cost and life time performances for the same building component. Assuming this, one can conclude that: i) a specification of a schedule of work is made up of different components; and ii) for the same ground of unfitness different schedules of work can be specified by modifying the type of work component. Figures 5.16, and 5.17 give an illustration of this problem.

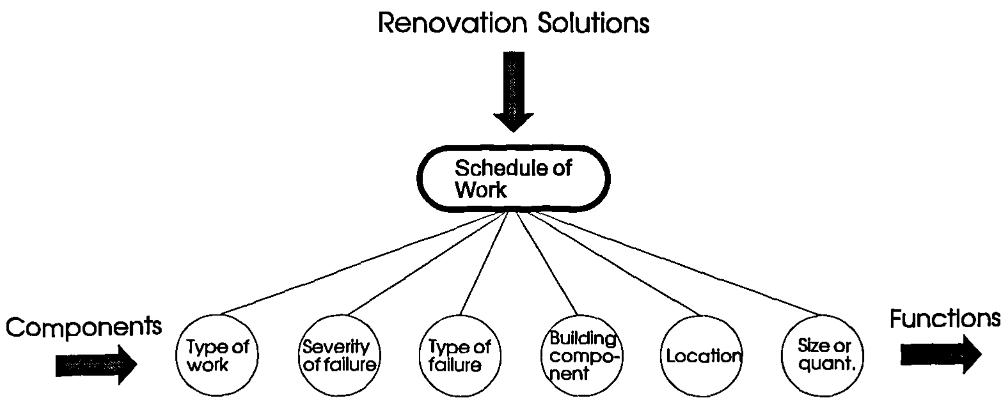


Figure 5.16: Components of the schedules of work

Thus, regarding the adaptation of schedules of work, what needs to be adapted is the type of work component. The rest of the components are given by the corresponding ground of unfitness, as can be seen in figure 5.18.

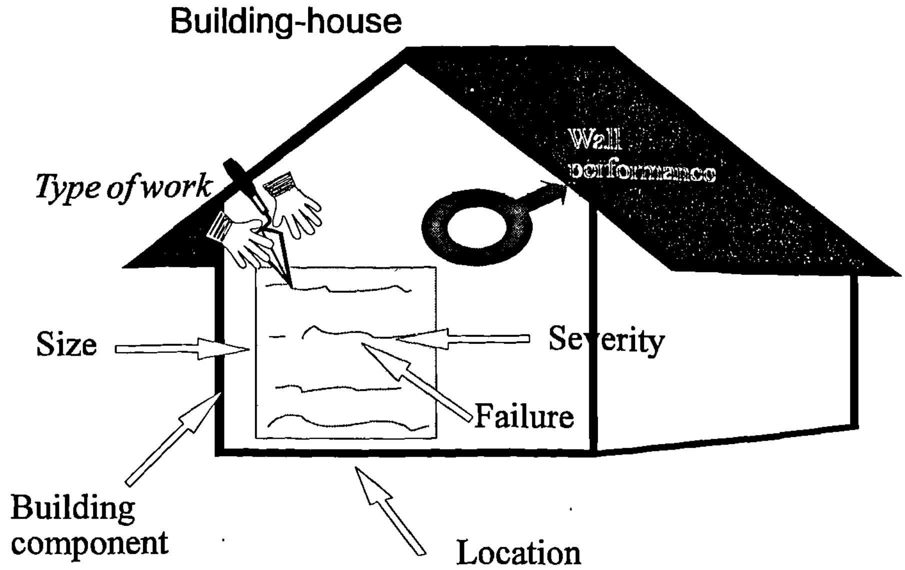


Figure 5.17: Illustration of a schedule of work

Past application cases store schedules of work by type of reason for unfitness. Because there are similarities of schedules of work within the same reason for unfitness, they can be used as a source for providing the types of possible adaptations.

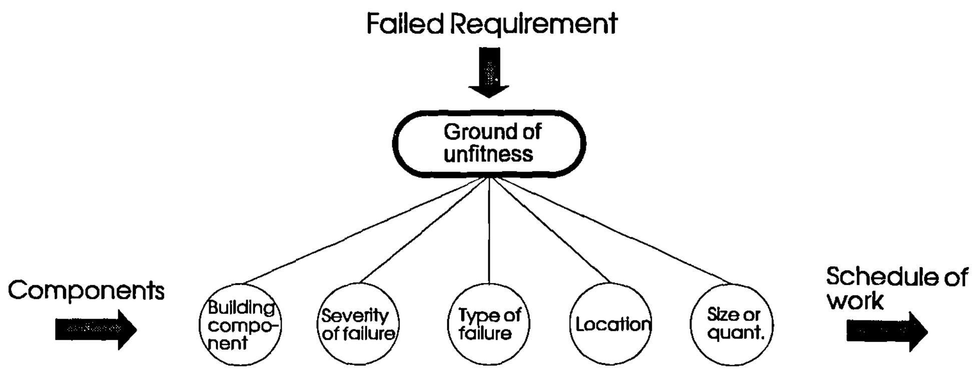


Figure 5.18: Illustration of the components of a ground of unfitness

The cost of schedules of work should include all costs which will be incurred in their execution. Two cost estimates are required to be provided by the applicant. These estimates will help to form a balanced view on the likely costs attributable to the eligible works. The attributable cost of eligible works should include: i) the cost of carrying out the works; ii) services and charges; and iii) VAT (where required).

5.5.4- KNOWLEDGE FOR UNDERTAKING THE ECONOMIC ANALYSIS

When an application is for a renovation grant and the property is deemed unfit, LAs must be of the opinion that renovation is the most satisfactory course of action to deal with an unfit dwelling-house. Thus, LAs are expected to make an economic assessment of the viable options for an unfit dwelling-house. The code of guidance for dealing with unfit premises recommends the use of an economic formula to assist in assessing the comparative financial merits of alternative courses of action in terms of their net present values (NPVs).

The formula includes total costs and benefits which can be described in monetary terms in order to ensure that decisions are based on an awareness of the overall economics of alternative courses of action. It enables a LA to consider costs and benefits over a 30 year period. The alternative options that might be taken into consideration in the economic appraisal are: i) renovation of the property; ii) demolition and rebuild; iii) demolition; and iv) maintaining the current state.

Costs and benefits that accrue to either the public or private sectors are included. Some of the costs and benefits expected to be included in the economic assessment are:

1. **Costs which might be included, are those for:**

- works required to make the property fit;
- works required over and above those to make the property fit but to secure its long term future;
- works required to maintain the value of the property if it is given a long term view;
- compensation which is paid to occupants where a demolition option is adopted;
- re-housing, where this is the duty of the local authority;
- securing a property which is closed or is to be demolished;
- demolition costs;
- work associated with retaining a vacant site; and
- new-build costs.

2. **Benefits to be included are**:

- increase in the market value of a property as a result of a renovation action; and
- increase in land values as a result of demolition or rebuilding.

To find the economic merit of a renovation action, its net present value has to be compared with the NPVs of the alternative courses of action available for the unfit dwelling-house. Also, the limit of £20,000.00 for the cost of the renovation option should be taken into account. The economic merit of a renovation option may fall into the following categories: i) Very good; ii) Good; and iii) Bad.

5.5.5- KNOWLEDGE FOR TESTING THE FINANCIAL RESOURCES

The test of financial resources enables a LA to work out how much the applicant(s) is (are) likely to be able to pay towards the cost of the works, i.e. the reduction in the amount of grant. If the *eligible income and capital* of the *relevant persons* exceed the *applicable amount* of those *relevant persons* then a *reduction in the amount of grant* is determined as a result of the means testing task.

A relevant person in respect of any application is a person who: i) is an applicant; ii) is not an applicant but is entitled to make an application and lives, or intends to live in the property; iii) where the application is for a disabled facilities grant, he or she is the disabled occupant; and iv) is a young person and he is the only applicant. The determination of who is a relevant person is illustrated by figure 5.19.

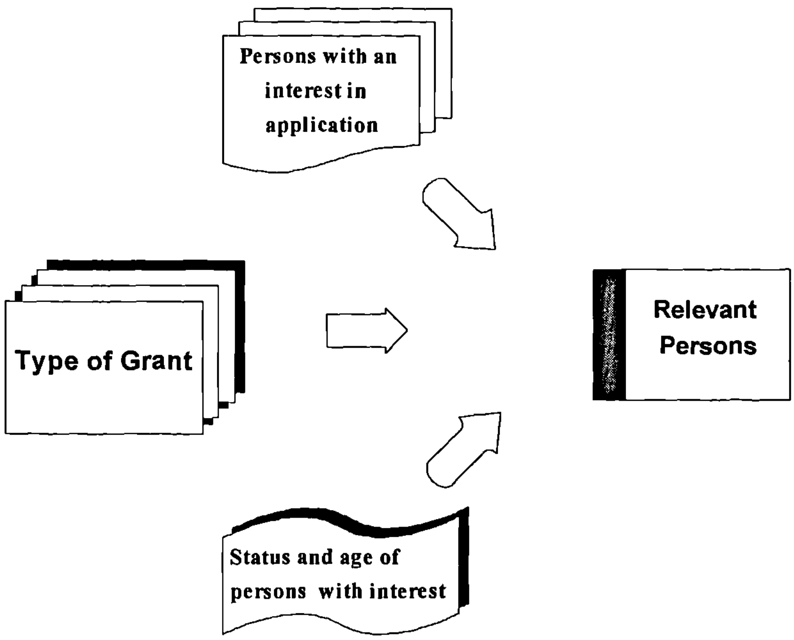


Figure 5.19: Procedure to find the relevant persons to an application

Each relevant person's applicable amount (the assessment of basic needs) is determined by the award of personal allowances in respect of the relevant person, his or her partner and any dependent children. A variety of premiums may also be awarded in respect of the special needs such as parenthood, old age or disability. The applicable amount in respect of any grant application is the aggregate of: i) the total weekly applicable amounts of all relevant persons; and ii) £40.00. The weekly applicable amount of a relevant person who is in receipt of income support is £1.00. The weekly applicable amount of a relevant person shall be the aggregate of one of the following, as may apply in his case: i) personal allowances; ii) family premium; and iii) premiums. The premiums are of the following types: i) lone parent premium; ii) pensioner premium for a person under 75; iii) pensioner premium for persons of 75 and over; iv) high pensioner premium; v) disability premium; vi) severe disability premium; vii) disabled child premium; and viii) carer premium. The procedure for determining the applicable amount is illustrated by figure 5.20. The income and capital of each relevant person, together with his or her partner, must be taken into account in the assessment of financial resources.

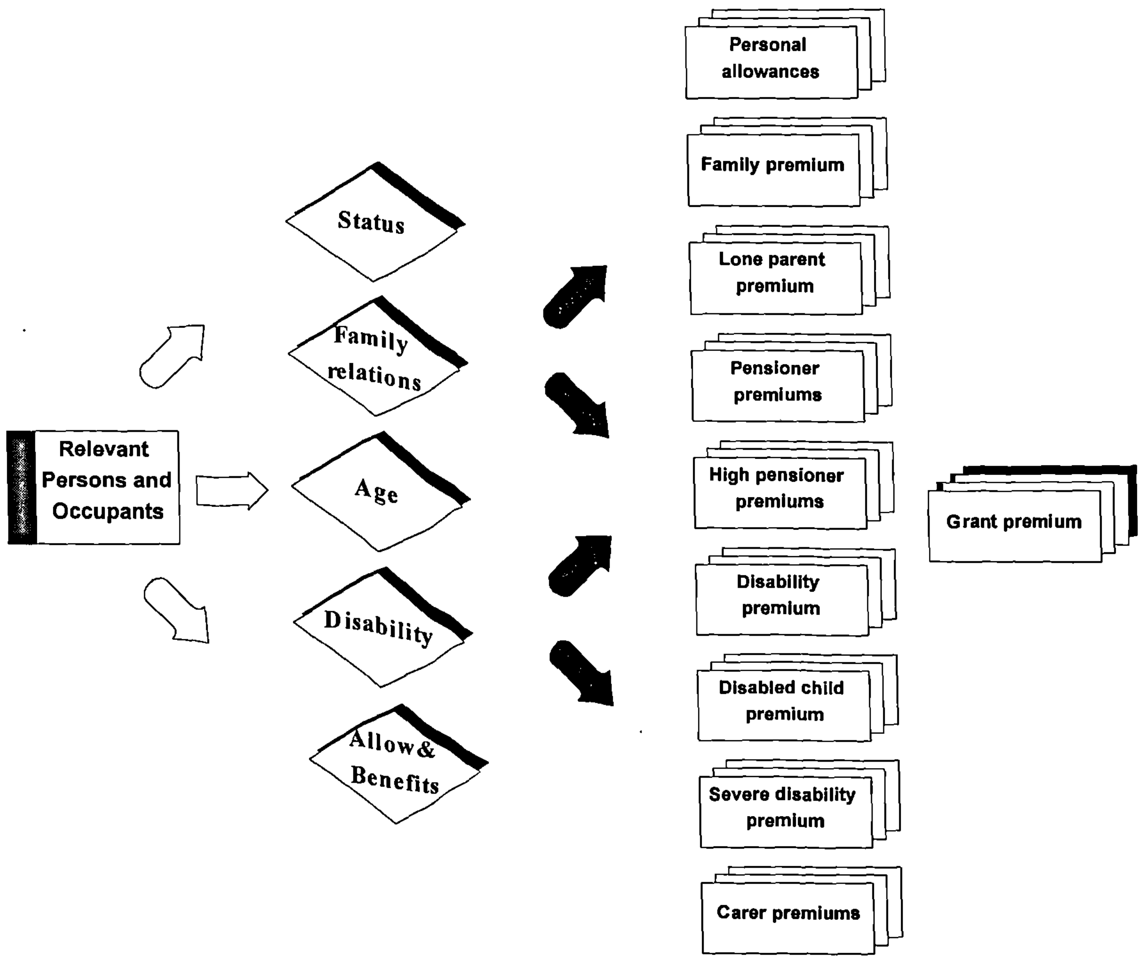


Figure 5.20: Procedure to find the applicable amount of the relevant persons to an application

In respect of an application, the amount that is to be taken as the financial resources of the relevant persons shall be the total of all the incomes of that person. The eligible income and capital of a relevant person shall be determined on a weekly basis by aggregating:

- his average weekly earnings from employment as an employed earner;
- his average weekly earnings from employment as a self-employed earner;
- his average weekly income other than earnings. Any part of a relevant person's income which does not consist of earnings; and
- the weekly tariff income derived from the capital held by the relevant person- where the relevant person's capital exceeds £5,000 it shall be treated as equivalent to a weekly tariff income of £1 for each complete £250 in excess of £5,000.

Where a relevant person, or his or her partner, is in receipt of income support, all their income and capital are completely disregarded. The procedure to calculate the eligible income and capital related to an application is illustrated by figure 5.21.

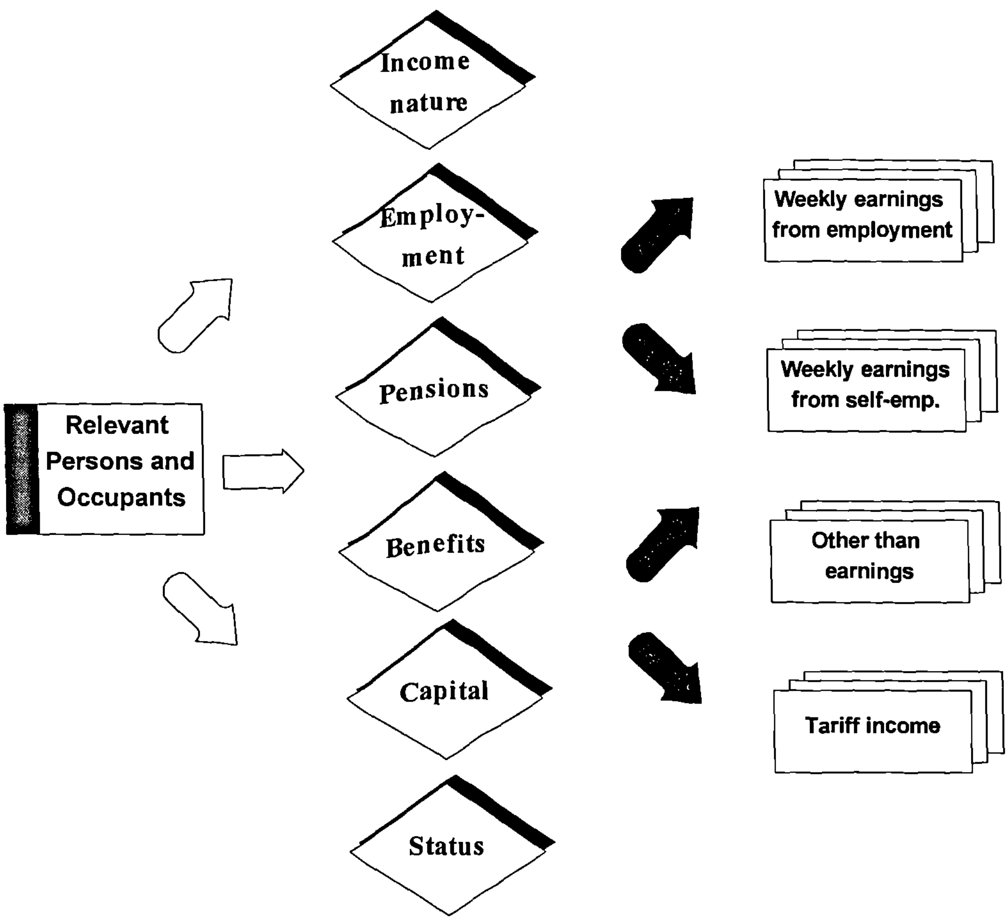


Figure 5.21: Procedure to find the income and capital of the relevant persons to an application

If the eligible income and capital exceed the applicable amount, in respect of any application, then the reduction in the amount of grant shall be calculated. If the applicant is on income support the reduction in the amount of grant is equal to 0.

5.5.6- KNOWLEDGE FOR DETERMINING THE AMOUNT OF GRANT

The amount of grant to be paid to a given applicant: i) where the total eligible income exceeds the applicable amount, is equal to the total cost of eligible works (including administration fees and charges) minus the reduction in the amount of grant; ii) where the total eligible income is less than the applicable amount, is equal to the total cost of eligible works.

The difference between the selected estimate and the calculated grant represents the part of costs shared by the applicant. Where a relevant person, or his or her partner, is in receipt of income support, 100% of the grant will automatically be available as long as all other relevant persons in the application are also in receipt of income support.

5.5.7- KNOWLEDGE FOR DETERMINING THE DEGREE OF ELIGIBILITY OF AN APPLICATION

On the basis of the experience provided by past applications, a classificatory framework was developed in this research to assist in assessing the general merits of a grant application. This tool helps to find a solution plan that meets the needs of that application. According to this framework, any application may fall into the following categories of eligibility for a grant: i) very high; ii) high; iii) medium; and iv) low.

The assessment of the eligibility of a grant application has regard to criteria. This criteria includes: i) the expected impact of completed works on the condition of the property; ii) the housing needs in the area; iii) the specific needs of the applicant and other occupants; iii) the results of the economic appraisal; iv) the level of condition of the property; v) the provision of a certificate of future occupation; and vi) past assistance received by the property. Figure 5.22 shows the procedure to evaluate the degree of eligibility of a current application.

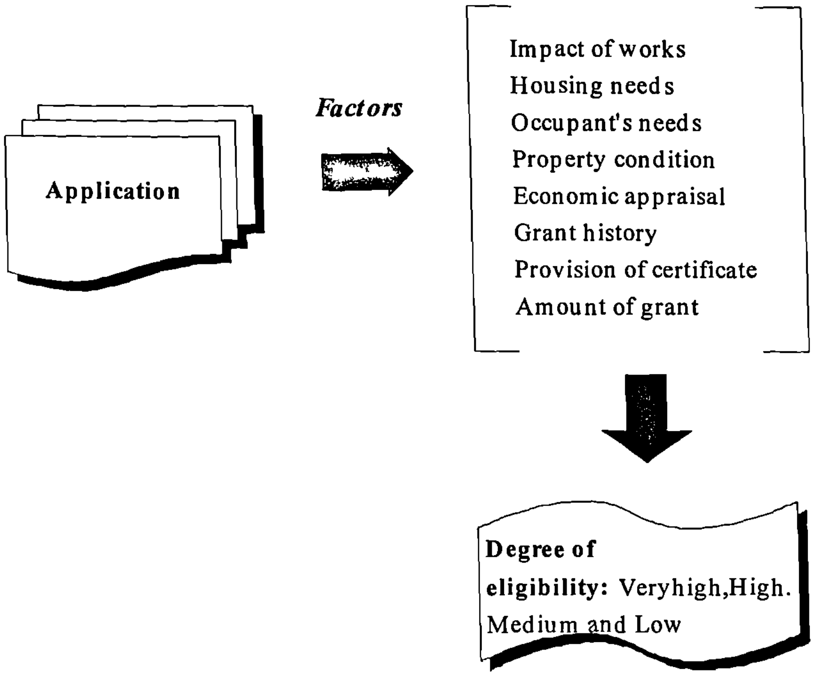


Figure 5.22: The procedure to evaluate the degree of eligibility of a current application.

5.5.8- HOW THE APPLICATION CASE SHOULD BE PRESENTED

The solution for a current application should be presented as a plan containing several pieces of information which specify the actions to be taken.

Appropriate information should be selected from the current application case which completes the solution plan. Additionally, the a right format to present that information should be adopted. Past applications provide an indication as to *what are the relevant pieces of information* and *what is the format* that should be used to present the solution plan.

The relevant information of a solution plan for a renovation grant should include the following: i) application code; ii) applicant name; iii) property address; iv) grant type; v) type of certificate provided; vi) fitness of the property regarding the standard of fitness; vii) reasons for unfitness; viii) schedules of renovation work; ix) condition of the execution of the works; x) means test result; xi) approved cost; xii) amount of grant; xiii) amount not granted; xiv) economic appraisal; xv) grant decision; xvi) condition of implementation of the grant; and xvii) recommendations.

The relevant information of a solution plan for a disabled facilities grant should include the following: i) application code; ii) applicant name; iii) property address; iv) grant type; v) type of certificate provided; vi) person registered as disabled; vii) fitness of the property regarding the standard of fitness; viii) welfare recommendation; ix) result of the assessment of the adaptation works; x) schedules of adaptation work; xi) condition of the execution of the works; xii) means test result; xiii) approved cost; xiv) amount of grant; xv) amount not granted; xvi) grant decision; xvii) condition of implementation of the grant; and xviii) recommendations.

The relevant information of a solution plan for minor works assistance should include the following: i) application code; ii) applicant name; iii) property address; iv) grant type; v) type of certificate provided; vi) schedules of adaptation work; vii) condition of the execution of the works; viii) approved cost; ix) amount of assistance; x) amount not granted; xi) grant decision; xii) condition of implementation of the grant; and xiii) recommendations.

5.6- SUMMARY AND CONCLUSIONS

In this chapter a general description of the knowledge required to solve the sub-tasks identified by the task structure has been outlined. It is not a complete discussion of all the knowledge represented in the system developed in the context of this research.

The case acquisition in the form of application cases has been an important step in all KA process discussed in this chapter. A number of practical benefits were obtained from acquiring knowledge from past applications and in the form of application cases.

The KA lead to several conclusions during the system's development cycle. Some of these conclusions are specifically related to the use of application cases to acquire the task knowledge. They can be summarised as follows:

- The task structure discussed in chapter 4 facilitated the acquisition of the task knowledge. It has strongly helped: i) to select the KA methods; ii) to identify what task knowledge to acquire; and iii) what sources of knowledge to use.
- Past application cases comprise a large body of knowledge which record expert's experiences in specific situations. They are records of how the Client expert's had applied their skills in practical contexts. These make past applications one of the major source of knowledge within the domain.

- The use of past application cases to acquire the task knowledge enables the filling of gaps in knowledge found in other sources of knowledge such as: the statutory and regulatory texts and technical publications.
- The use of several past application cases to acquire task knowledge from Client's experts seemed to be a very useful method. Client's experts felt more motivated to provide their knowledge in cases they have processed in the past. They were able to remember experiences they have encountered in specific situations. Therefore, this method allowed the more efficient use of the limited time available from the Client's experts.
- The KA was facilitated by acquiring task knowledge in terms of application cases. For Xtimela-CBR, even an initial, incomplete case library of only 60 cases provided a product that the Client found potentially useful. Also, this enabled the system validation at an early implementation stage.

In the next chapter an architecture which underlies the task structure discussed in chapter 4 and embodies the knowledge described in this chapter will be discussed. Additionally a computer tool for implementing the system will be introduced.

CHAPTER 6

SYSTEM'S ARCHITECTURE

6.1- INTRODUCTION

This chapter describes an architecture for the system proposed in this research. This architecture is designed to organise the knowledge types described in chapters 4 and 5 effectively within the system and to support various processes identified by the task analysis. It describes the system's components needed for representing and using the knowledge required by the assessment of grant applications tasks. This architecture was used to implement the system as a computer application using the IntelliCorp's Kappa-PC© shell. This computer application was called Xtimela-CBR.

This chapter consists of two parts. The first part, introduces issues related to the architecture and presents an overview of its basic components. The second part, introduces Kappa-PC shell and presents the reasons for choosing it as implementation tool.

The benefits and weaknesses of the architecture presented in this chapter are discussed in chapter 7.

6.2- ARCHITECTURE FOR THE SYSTEM

6.2.1- SOME ISSUES RELATED TO THE ARCHITECTURE

Brandon (Brandon P., 1993a) pointed out four immediate challenges for KBSs in the construction industry. They are to:

- Improve the input problem.
- Enhance output support.
- Support conventional modelling.
- Aid information retrieval and support for decision makers.

An architecture which addresses these challenges could lead to the development of useful and efficient KBSs. Systems in use and in development such as, ELSIE, EMMY and CASHFLOW are practical examples which prove the effectiveness of those four challenges mentioned above. How these challenges can be addressed by the system proposed in this research was a key issue in designing the system's architecture.

The review of KBSs using CBR presented in chapter 3 has indicated four major types of architectures that have been used so far. They are:

- purely CBR architectures (CHEF);
- architectures that incorporate other methods in support of CBR (JULIA, KRITIK, ARCHIE, CADSYN);
- architectures that combine CBR with other methods (CABARET, CYCLOPS); and
- architectures where CBR comes in support of other methods (CASEY).

Therefore, the review has shown that architectures employed by systems using CBR were often related to: i) the tasks which the system was designed for; ii) the processes incorporated in the system; and iii) the purposes of cases in the system when they are recalled from the case library.

Regarding to the level of automation, some architectures were designed for full automated systems. Others were designed for interactive aiding systems. The interactive aiding systems can provide varying degrees of aid to a human user. At its simplest, an aiding system can act only as a case retriever, by providing cases to a user who use them for tasks such as: interpretation, evaluation, advising and diagnosing. More sophisticated systems using CBR combine retrieval, evaluation and adaptation of cases stored in the system for supporting a number of problem solving tasks such as, design and planning (Kolodner J., 1993).

Smith and Johnson (Smith J. and Johnson T., 1993) pointed out that knowledge-level descriptions can provide a basis for designing KBS's architectures. The three versions of the task structure developed for the assessment of applications for HRGS underlie an architecture for Xtimela-CBR.

6.2.2- OVERVIEW OF SYSTEM'S ARCHITECTURE

One of the issues of this research was to design an architecture for the Xtimela-CBR system that:

- integrates CBR with other reasoning methods as specified by the task structure;
- allows to represent within the system different type of knowledge required by the overall task;
- integrates a memory of cases, problem solving and learning into one framework;
- supports modular implementation of the system;
- makes inferences more apparent to the user by providing clear explanations where appropriate;
- facilitates knowledge encoding and maintenance over time; and
- is able to operate for the following versions of the domain: i) assessment of application for renovation grant; ii) assessment of applications for disabled facilities grant; and iii) assessment of applications for minor works assistance;
- can be implemented in a low cost computer shell.

Taking into account these requirements and on the basis of the review and guidelines outlined in chapter 3 an integrative architecture was designed for Xtimela-CBR system. An overview of this architecture is presented in figure 6.1.

Figure 6.1 shows the basic components of the system's architecture and highlights the relationships between them.

The Xtimela-CBR architecture is independent of the implementation shell and was designed to work in the three versions of the domain defined in the scope of this research. It consists of eight main components, which are described in the following sections.

Xtimela - CBR architecture

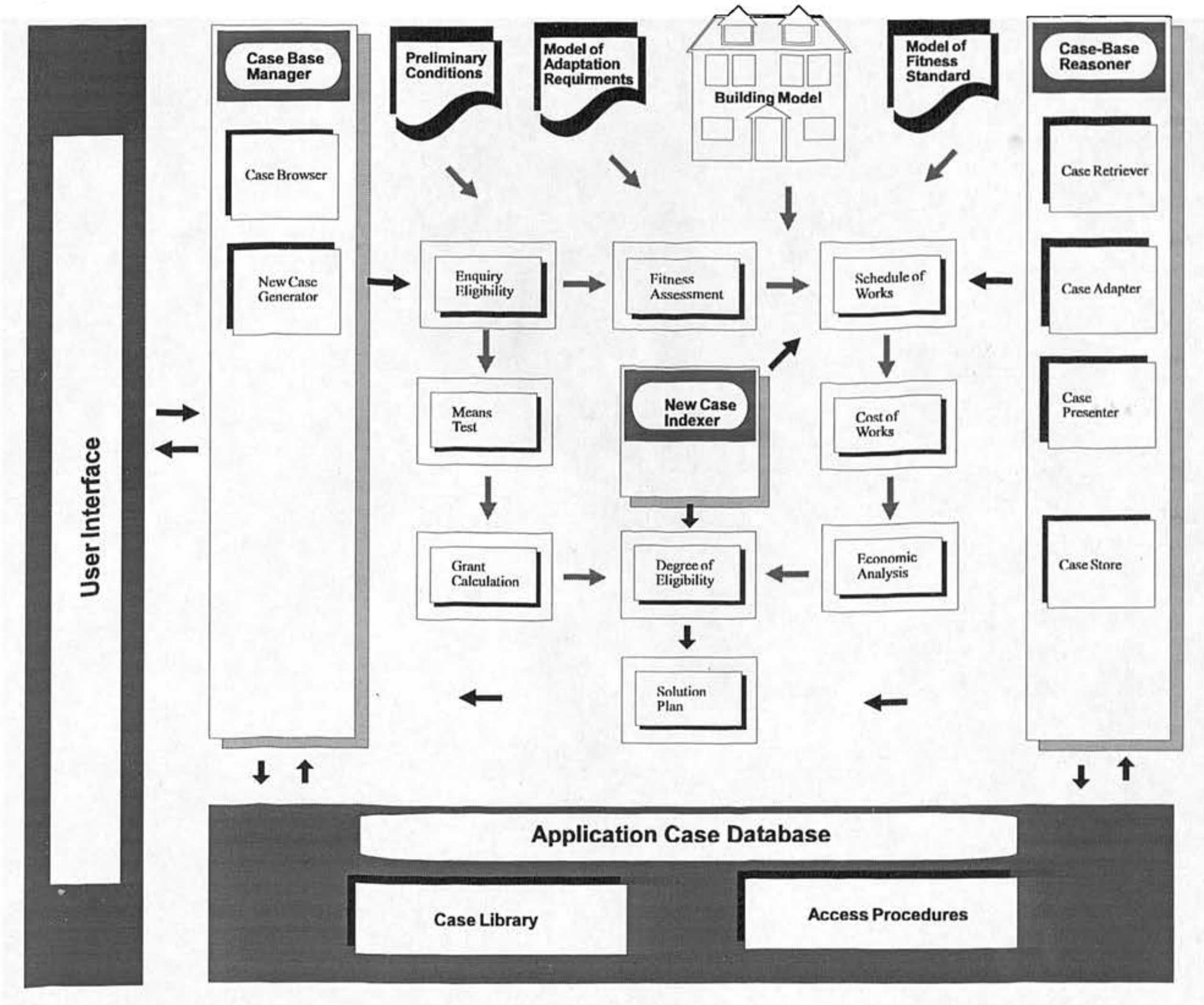


Figure 6.1: The architecture for the system proposed in this research

6.2.2.1- The Application Case Database

Kolodner (Kolodner J., 1993), pointed out that, a system using CBR can only be good as its memory of cases. This statement emphasises the crucial importance of the application case database component in the Xtimela-CBR architecture. Part of the knowledge required by the system is stored in the application case database in the form of application cases. The application case database was designed to allow the system to perform the following functions:

- to organise, store and index application cases within the system;
- to provide a case representation structure for a current application; and
- to make application cases stored in the case library accessible for problem solving and consultation purposes.

To perform the functions mentioned above the application case database incorporates four main sub components. They are:

- **the case library** which comprises a representative set of successful past application cases; and
- **a set of *procedures*** to access the case library.

Each application case in the case library represents the most relevant information contained in a past application. Application cases in the case library contain a set of indexes which designate in what circumstances a relevant past application case should be retrieved.

The application cases stored in the case library are of the following types corresponding to the scope of this research:

- renovation grant application cases;
- disabled facilities grant application cases; and
- minor works assistance application cases.

6.2.2.2- The New Case Indexer

The case indexer was designed to allow the system to perform the following functions:

- to assess the current situation regarding the application being processed; and
- to select the relevant indices for the current task where the CBR method is invoked.

The indices with their values provide the retrieval cues for:

- guiding the search in the case library;
- assessing the similarity between the current specification and the specification of the potentially similar case; and
- matching with potentially similar application cases stored in the case library.

6.2.2.3- The Case-Base Reasoner

The case-base reasoner is invoked by the system's control for finding: i) the schedules of works required to make fit an unfit dwelling-house; and ii) a decision for the current application case. The case-base reasoner was designed to allow the system to perform the following functions:

- given a set of indexes, to searche the case library to find a potentially similar application case containing a solution to the current problem;

- to find the correspondence between the specification of the current problem with the specification of those solutions stored in the potential similar case;
- to compute the similarity of corresponding specification features;
- to retrieve a similar application case from the case library;
- where the case retriever fails to find similar application case, to adapt an old solution in order to meet the current specification;
- to present to the user the solution plan for the current application case; and
- to store a current application case in the case library for future retrieval.

To perform these functions the case-base reasoner incorporates four main sub-components:

- **the case-base retriever**: allows the system to retrieve relevant cases given a description of the current problem;
- **the case-base adapter:** which is called to adapt a schedule of work for an unfit building component;
- **the case reporter:** allows the system to present to the user the solution plan found for the current application; and
- **the case storage:** allows the system to store an application case in the case library after has been processed.

6.2.2.4- The Case-Base Manager

The case-base manager was designed to allow the user to access the application cases stored in the case library. To perform this function the case-base manager incorporates two sub-components as follows:

- **the case browser:** for browsing, modifying and displaying cases stored in the application case library when required by the user; and

- **the case generator:** for creating a new application case, which refers to a current application.

6.2.2.5- The User Interface

User interface design is critical for attaining wide-spread system acceptance in the Client's organisation. A well designed graphical user interface makes a system's purposes and conclusions more accessible to the users (Irgon A. et al., 1990).

Xtimela-CBR has an interactive architecture which incorporates a window-based graphical user interface. This user interface was designed to allow the system:

- to provide the user with access to the case-base manager, new case indexer, and the case-base reasoner;
- to allow the user to maintain and modify the information stored in the system; and
- to provide the user with access to the control of the system.

6.2.2.6- Domain Models

Domain models in the system represent the static knowledge which is required to perform the assessment of grant application tasks using the problem solving methods described by the task structure. Taking into consideration the scope of this research the current version of Xtimela-CBR, uses four types of models:

The Building Model:
Because this research focuses on dwelling-houses only, the building model expresses the physical structure of a dwelling-house and interrelationships among its building components. The building model captures the knowledge about:

- ♦ the building components of a dwelling-house commonly found in past applications; and
- ♦ the interactions among those building components.

Thus, the building model comprises a collection of building components, each one performing a function within the house, alone or together with other neighbour components. Associated with each building component are features which describe them in terms of the knowledge required by the system. According to the task structure the building model was designed to support the following task:

- Assessment of the fitness of the property.
- To index a new case application.
- Finding the schedule of works for an unfit house.

The Standard of Fitness Model:
The standard of fitness expresses the relationship between the requirements of the fitness standard for human habitation and building components of a typical dwelling-house. The standard of fitness model captures the knowledge about:

- ♦ the requirements of the fitness standard for human habitation associated with each building component;
- ♦ the interrelations among the various requirements;

- the usual procedures for assessing the fitness of a dwelling-house;
- the typical failures to the fitness standard associated with each building component of the building model; and
- the possible grounds of unfitness associated with each building component of the building model.

The standard of fitness model supports the task of assessing the fitness of a property.

The Grant Conditions Model:
The grant conditions model expresses the preliminary conditions which an enquiry for a grant application must meet. This model captures the knowledge about: i) the conditions of eligibility for the HRGS; and ii) the procedures for evaluating an enquiry for a grant application.

The Disabled Facilities Model:
The disabled facilities model expresses the relation between the legal requirements and the disabled facilities adaptations available for a typical dwelling-house. It captures knowledge about:

- the legal requirements;
- the disabled facilities adaptations;
- the procedures to assess if the adaptations are necessary and appropriate; and
- the procedures to assess if the adaptations are reasonable and practicable attending the condition and age of the dwelling-house.

6.2.2.7- The Control Of Processing

With appropriate control of processing knowledge, the system will know directly which task to take at each step of the problem solving of assessing a given grant application so it can reach a goal state (Smith J. and Johnson T., 1993). To achieve this, the Xtimela-CBR architecture incorporates three different mechanisms for control of processing: i) the top-level control; ii) the intermediate level control; and iii) the local-level control. Control of processing means the selection of the operations to be performed at any given stage of the problem solving task (Goel A., 1989). Figure 9.2 illustrates these mechanisms of control. A description of each mechanism is given in the next paragraphs.

Top-level control of processing:
The top-level control of processing is *task-directed*, i.e., the task structure suggests which task to perform next. For example, the task of finding the schedules of work is triggered whenever the results of the fitness assessment are made available. This task-directed control

mechanism is static and rigid since it is encoded (i.e., built-in) into the Xtimela-CBR system.

Intermediate-level control of processing:

At the next lower level, specific tasks are used to guide the selection of particular methods for accomplishing them. For example, the means testing task is used to select the method applicable to it. Goel (Goel A., 1989) called this task-specific selection of methods by *instance-specific selection of method.* This task-specific selection of method control is rigid since it is encoded into the Xtimela-CBR system.

Local-level control of processing:

The local-level control of processing is method-specific, i.e., the problem solving methods suggests which sub task to perform next. At this level, each particular problem solving method for accomplishing a given task is described in the task structure with a control which is specific to it. This method specific control mechanism include sub-tasks and search control knowledge for sequencing the sub tasks (as described in chapter 4). For example, the CBR method performs the task of finding the schedules of works by decomposing it into three sub tasks and sequencing their implementation.

A description of a problem solving session with Xtimela-CBR for assessing an application for a renovation grant illustrating the system's overall control of processing is presented in Appendix 10.

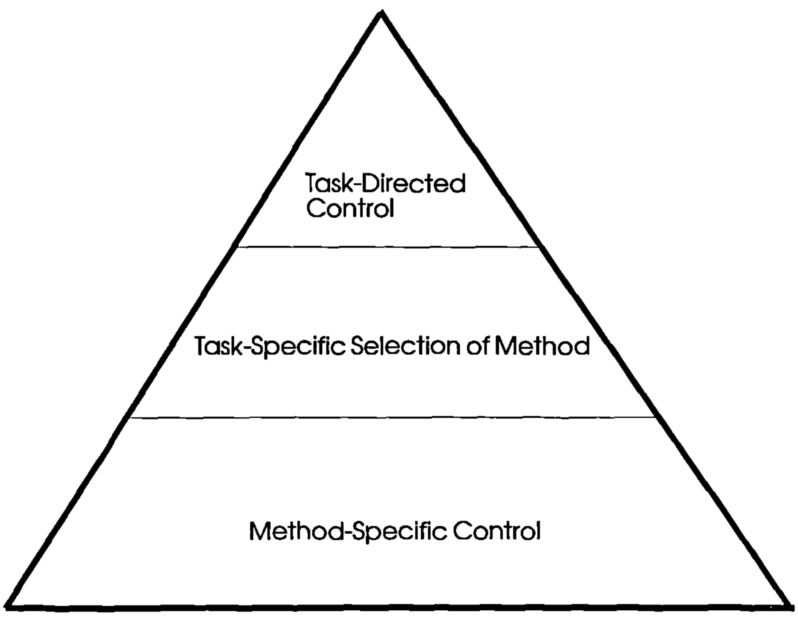


Figure 6.2: The illustration of the system's overall control of processing

6.2.2.8- The Explanation

For a KBS to be accepted, it must firstly be accountable. An accountable KBS must be able to provide explanations for its reasoning and justify its conclusions (Motoda H. 1994). Chandrasekaran's (Chandrasekaran B. et al., 1992) task structure uses task-specific explanation modules for explaining tasks at an abstract level. These modules represent the content which a task-specific explanation must convey. Thus, Xtimela-CBR architecture incorporates explanation modules as described in the task structure.

Xtimela-CBR explanation modules were designed to perform three types of functions. They are:

- relate the system's actions and conclusions to the goal of the task it performs; and
- relate decisions to particular lines of reasoning.

6.3- KBS TOOL FOR THE IMPLEMENTATION OF Xtimela-CBR

6.3.1- KBS TOOLS

Although there are a number of KBS development tools which can be used to implement the system using the architecture presented in section 6.2, some of them can make the implementation much easier than others.

A KBS development tool, is a software development environment containing basic components of KBSs. Associated with a KBS development tool is or are prescribed method(s) for building real-world applications. In recent years, new types of tools have come on the market that are specialised according to task (diagnostic and design), problem solving method(s) (CBR, model-based reasoning, rule-based reasoning), or development methodology (Feigenbaum E. et al., 1994).

The choice of an adequate KBS development tool for implementing Xtimela-CBR was made considering a number of more general requirements. Thus, a KBS development tool for implementing the system should:

1. Provide means for representing, indexing and organising past applications cases into a case library.
2. Provide means for representing other knowledge types required by the system such as: domain models, procedures and search and control knowledge.
3. Provide tools for implementing the components of the system's architecture.

4. Provide tools for developing and customising the user interface to the system.
5. Provide tools for developing and customising explanations.
6. Provide tools for debugging and verifying the program.
7. Provide text-matching algorithms.
8. Facilitate the maintenance over time of the knowledge represented in the system.
9. Be easy to use for development and running purposes.
10. Not be expensive.
11. Run in a personal computer (PC) computer equipped with a processor 80386 category PC/AT.

At the time when the research started, four KBS development tools available on the market were considered as candidates for implementing the system. They were: i) Kappa-PC version 2.1 from IntelliCorp, Inc.; ii) CBR Express version 1.1 from Inference Corporation; iii) ART-IM version 3.0 from Inference Corporation; and iv) ReMind from Cognitive Systems, Inc.

6.3.2- BRIEF DESCRIPTION OF THE CANDIDATE KBS DEVELOPMENT TOOLS

This section presents the results of an evaluation of four KBS development tools potentially useful for implementing Xtimela-CBR. The goals of this study were:

- to assess the functionality and performance of tools with different architectural and reasoning approaches; and
- to gain insight into the advantages of each tool for implementing the system.

The KBS development tools were evaluated using a criteria that measure issues in the following categories: i) knowledge representation; ii) reasoning processes; iii) development environment; and iv) technical support and price.

6.3.2.1- KAPPA-PC

Kappa-PC is a "personal computer"(PC) KBS development tool with object-oriented programming capability based on the basic ideas of objects and object inheritance. It is a generic tool that provides means for development of KBS applications for several reasoning tasks, such as: i) diagnosis and classification; ii) design and configuration; iii) planning and scheduling; and iv) simulation and process control. However, Kappa-PC is not a tool specifically featured for the development of CBR systems. It does not incorporate any of the built-in mechanisms provided by CBR tools such as: i) ReMind; and ii) CBR Express.

Knowledge Representation

Object-oriented programming (objects, classes, instances and inheritance), rules and functions are the basic building blocks of Kappa-PC knowledge bases.

Object-oriented programming provides means for representing the behaviour of individual objects. Objects are the means for representing entities in a KBS which can be related to each other to capture the relationships between the things or concepts they represent. In order to tailor an object so that it represents all of the important properties and behaviour of the entity, *slots* (features) and *methods* can be given to that object. Each slot describes a characteristic of the object. Each method describes the behaviour of that object. Objects in Kappa-PC can:

- have their own slots and methods; or
- inherit slots and methods from ancestors; or
- share slots and communicate with each other using a method.

Objects are defined as either *classes* or *instances*. A *class* is a more general object; it can be a group or collection. An *instance* is more specific object; it is a particular item or event. Objects in an application can form one or more hierarchies of *super-classes*, *subclasses*, and *instances*. Organising objects into a hierarchy allows the inherent structure of a knowledge base to be represented explicitly. Hierarchies can relate to each other through object network links.

Rules in Kappa-PC are used to represent *if-then* reasoning, which may or may not relate to individual objects in the knowledge-base. Rules can be built as rule sets or incorporated into methods and functions.

Functions are a powerful tool in Kappa-PC language. Kappa-PC provides a library of over 240 functions. These functions can be used to define expressions or other more complex functions.

Reasoning Processes

Kappa-PC provides both forward and backward chaining reasoning capabilities. Forward or event-driven reasoning is used to determine the consequences of a new fact, and the consequences of the consequences. Backward or goal-driven is used to pose a question to be answered (goal to be established). The question can prompt the chaining- the asking of secondary questions.

Development Environment

Kappa-PC provides a flexible environment that integrates a set of basic programming tools which, with additional programming work, can be used for implementing and testing the system proposed in this research. Kappa-PC development environment allows a developer to choose from a variety of means of developing a KBS application. These means include (IntelliCorp, 1992):

1. Object-oriented programming features: i) the ability to represent, organise and process knowledge using *classes*, *instances*, *objects*, *methods* and *functions*; and ii) the ability to organise related objects into hierarchies using *inheritance* and *methods*.
2. Rule-based reasoning: *rules* in Kappa-PC have a standard *if-then* form and are manipulated by an inference engine, which manages the chaining of rules by linking up the premises of one rule with the conclusions of another.
3. A programming language called KAL that allows a programmer to create specific tailored functions and methods.
4. A graphical user interface that provides: i) a set of editor tools which facilitate the fast development of prototype applications; and ii) a set of object images that allows one to build easy-to-use window-based user interfaces. The windows and object images are the basic component of Kappa-PC user interfaces tools.
5. A set of functions for manipulating strings and lists.

Technical Support and Price

Kappa-PC has been used by the Department of Surveying for developing several KBSs. The Department of Surveying has accumulated valuable experience as result of working with Kappa-PC and IntelliCorp. Thus, technical support was readily available. Therefore, for the purpose of using Kappa-PC for the present research it would require only to purchase an additional licence.

6.3.2.2- ART-IM

ART-IM version 3.0 was designed for personal computers running under MS-DOS operating system and MS-Windows. It is a C-based tool that incorporates a sophisticated programming environment. It is generic tool which supports the development of KBS applications for several reasoning tasks. ART-IM is one of the most powerful commercial tools for PCs available today.

Knowledge Representation

ART-IM supports object-oriented programming as an extension of its schema system and provides strong data-driven rule-based reasoning capabilities.

An ART-IM object is represented by a schema consisting of a schema name and one or more slots. Slots represent individual items of information describing either attributes of the schema or its relationship with other schemata (functions to carry out object's actions). Procedural knowledge (user written functions) can be attached to schemas. Messages can be sent to an object using the *send* command.

ART-IM provides single inheritance scheme in which both slot values and functions are inherited via *is-a* and *instance-of* relations between schemas. A schema that contains the *is-a* relation can have children schemata, whereas a schema that contains the *instance-of* relation can not have descendants. All schemas and slots are visible and freely accessible to rules and procedures. Rules represent *if-then* reasoning and can be related to schemas in the knowledge-base. Functions can be written either in C language or using a set of ART-IM commands.

Reasoning Processes

ART-IM provides a strong forward-chaining capability based on the recognise-act cycle and the RETE matching algorithm. ART-IM provides reasoning capabilities such as conflict-resolution strategies. Functions written in C or with ART-IM commands can be used to access schemas in the knowledge-base. ART-IM provides three text-matching algorithms such as: string, word, and character.

Development Environment

ART-IM development environment includes the knowledge base builder and the inference system builder. The knowledge base builder provides a graphical interface that includes options such as pop-up menus, windows and a cut-and-paste feature that allows users to build a knowledge base with click-and-pointing mouse actions. ART-IM includes a match feature that allows the user to test the application as each rule is being defined. Therefore, ART-IM provides a set of debugging aids.

Technical Support

Technical support was available from Inference Europe Ltd at a negotiable price.

6.3.2.3- CBR Express

CBR Express is client-server application shell of ART-IM. CBR Express, which runs under Microsoft Windows environment, is based on case-based reasoning. It is a purely CBR development environment.

Knowledge Representation

CBR Express provides a case representation structure consisting of one or more fields describing pieces of specific information. Within these fields can be attached pointers to additional information, graphics, on-line databases and associated documents. Cases reside in the configuration-controlled case-base.

A weight can be associated to each case field to indicate its importance in determining the closeness of a match. Similarly, a weight can be assigned to each question that specifies the importance of a question's answer. The overall matching score is defined by the combination of these weights during the search process.

Reasoning Processes

CBR Express automates problem solving using some of the typical CBR reasoning processes. Once the case data have been entered, the user goes to the Search Case Base screen to initiate the CBR Express reasoning engine. Based on the user's entry of the problem description, CBR Express searches the case base to find candidate cases that have some degree of match to the current case description. One problem with the weighting scheme used by CBR Express was that if an important question or description was input incorrectly by a user, it could dominate, and the correct matching solution might not be found. CBR Express does not provide means to carry out the adaptation of a past solution.

Integration with ART-IM is required for expanding the CBR Express reasoning capabilities.

Development Environment

CBR Express provides a development environment that allows end users to dynamically create and manage a case-base of past cases on their own in a production environment without assistance from an experienced programmer. With natural language support, CBR Express accepts and processes any information that is provided by a user in everyday English, even when some information is missing.

Technical Support

Technical support was available from Inference Europe Ltd at a negotiable price.

6.3.2.4- ReMind

ReMind is a generic tool designed for developing and deploying of CBR applications. It provides facilities for development of CBR applications for several reasoning tasks such as: prediction, classification, diagnosis, design and help desk. ReMind runs on a variety of hardware platforms. It is a CBR development environment.

Knowledge Representation
ReMind provides facilities for designing a case representation consisting of one or more features, entering case information, setting up a retrieval mechanism, designing qualitative models, supplying heuristics for adaptation and browsing through retrieved cases. Case features can be of several types such as: concepts, text, integers, real numbers, or functions. ReMind provides capability to augment case representation with domain models. Cases can be organised into clusters sharing common characteristics and/or flat memory.

Reasoning Processes
The indexing of cases in ReMind is based on: i) conceptual categories of cases defined by an user; ii) abstract derived features; and iii) user-supplied assessment of importance.

ReMind provides two algorithms for retrieving candidate cases: decision-tree traversal and/or nearest-neighbour. The matching process is based on nearest-neighbor algorithm. This algorithm computes the nearest-neighbor match score for each candidate case retrieved from the case library. ReMind provides capability for case adaptation using expert-supplied heuristics.

Development Environment
ReMind provides facilities for rapid prototyping of CBR applications. Thus, it provides graphical editor facilities for: i) importing data from existing databases; ii) designing a case representation iii) designing customs forms for the display of cases; iv) building semantic hierarchies defining abstractions over symbolic data; v) browsing through retrieved cases; and vi) dynamically computing case features. A C-function library that allows integration with other systems is provided by ReMind.

Technical Support
Technical support was available from Cognitive Systems, Inc.

6.3.3- SELECTION OF THE IMPLEMENTATION TOOL

The task analysis described in chapter 4 has indicated that the assessment of grant applications is a complex activity consisting of several tasks which are accomplished using different problem solving approaches. Five types of problem solving methods were identified for accomplishing the different tasks of the assessment of applications for the HRGS. This nature of the HRGS domain let the author to discard the use of any purely CBR development tool. CBR Express and ReMind although they have strong CBR capabilities they do not provide alone the reasoning approaches required to implement the system proposed in this research. Thus, CBR Express and ReMind were discarded from the candidate list.

ART-IM combined with CBR Express provided a very powerful environment for implementing the system. Their capabilities could not be fully used with the application proposed in this research. Therefore, the research was constrained by a set of limitations including money and time.

Kappa-PC, although was less powerful than ART-IM and CBR Express, provided a flexible development environment and several means which were considered potentially enough for implementing the system proposed in this research. Therefore, technical support for Kappa-PC was readily available in the Department of Surveying. The price to purchase a Kappa-PC license for the purpose of this research was substantial less than the price for purchasing ART-IM plus CBR Express.

Taking into account the general requirements set at beginning of section 6.3 and the discussion presented above, Kappa-PC version 2.1 was chosen as the implementation tool, and Microsoft Windows and PC computer were chosen as the implementation platform. Windows was chosen for the facilities offered by that version of Kappa-PC. Kappa-PC *Objects*, *object hierarchies*, *inheritance*, and *functions* are the main building blocks of Xtimela-CBR code. They were used to build the components of the Xtimela-CBR's architecture presented in section 6.2.2. Using the basic tools provided by Kappa-PC, in fact implied more programming work in building the system. On other hand, using Kappa-PC allowed for the better understanding of the concepts and meanings of the CBR models.

6.4- SUMMARY AND CONCLUSIONS

In the first part of this chapter an architecture for Xtimela-CBR system has been discussed. In the second part, four KBS development tools were introduced and the reasons for choosing Kappa-PC to implement the system were presented.

The architecture presented in this chapter was designed to explore the viability and validity of the CBR models in taking advantage of the large body of specific knowledge recorded in hundreds of past applications available in the Client offices. This architecture describes the components required for representing and using the knowledge needed to perform the assessment of grant applications tasks.

The architecture described in this chapter is a natural consequence of the task analysis discussed in chapter 4 which produced a specification for the system. Three key advantages are provided by this architecture:

- It supports modular implementation of the system because knowledge may be added or removed without having to alter the knowledge-base structure. Therefore, modularity will facilitate future maintenance of the system.
- It supports the representation of different types of knowledge needed for performing the assessment of grant applications tasks.
- It provides the user with an interface to control the execution of the system.

Although the architecture developed in this research is expected to provide some benefits for the system, it has the potential of being improved in terms of the functions it can accomplish.

Kappa-PC was chosen to implement the system using the architecture discussed in the first part of this chapter. It provides programming tools which have been tested in past research projects in the area of KBSs. Therefore, the Department of Surveying has accumulated a valuable experience using Kappa-PC for developing KBSs.

The next chapter consists of a description of the implementation of the system and of its main features.

CHAPTER 7

IMPLEMENTATION OF THE SYSTEM

7.1- INTRODUCTION

In chapter 6, an architecture and a tool for implementing the system have been discussed. The architecture describes the system's basic components including their functions. The implementation tool provides programming means for encoding system's components as a computer application.

This chapter presents a general description of the Xtimela-CBR working system. Therefore, it illustrates how some of the most important system's components were implemented using Kappa-PC shell. Figure 7.1 illustrates the implementation of the system as a natural consequence of the modelling work outlined and discussed in previous chapters combined with a technology for developing KBSs.

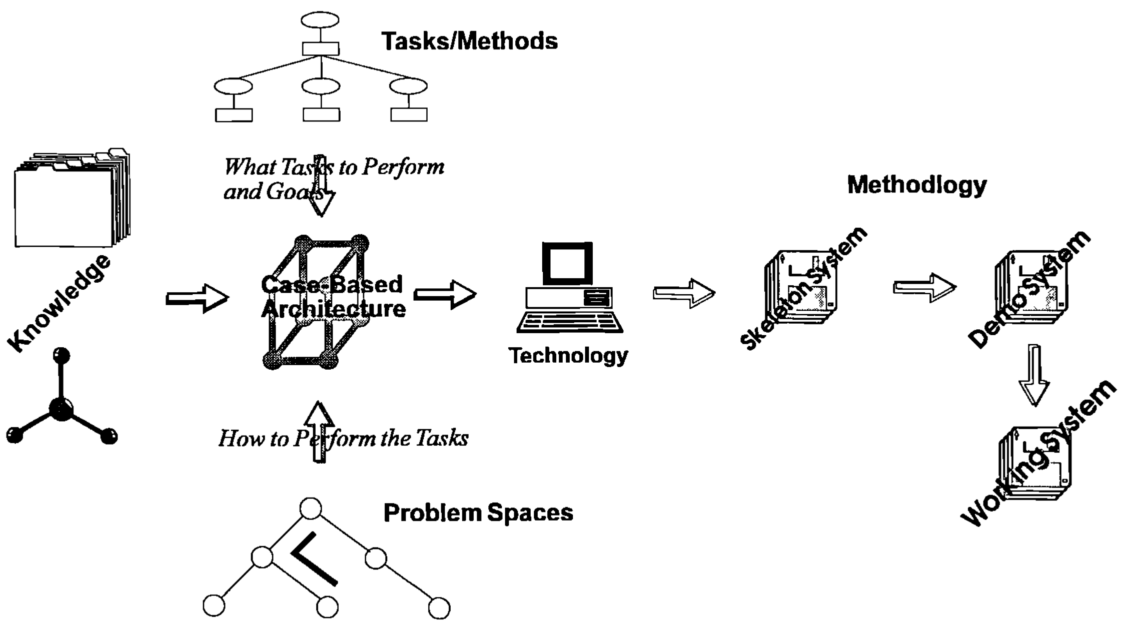


Figure 7.1: The implementation process of the Xtimela-CBR working system

Xtimela-CBR system, has a size of 1,256,394 bytes in KAL format at the present implementation stage and it runs on PC's under Microsoft Windows©. It stores 155 past application cases in the case library. Rules used by Xtimela-CBR are incorporated into its functions and methods. This system was implemented for assessing applications for: i) renovation grants; ii) disabled facilities grants; and iii) minor works assistance. All of the applications cases stored in the Xtimela-CBR application case database have been provided by the Client and represent real situations.

Majority of the figures presented in this chapter are based on Xtimela-CBR screens.

7.2- IMPLEMENTATION OF THE SYSTEM

7.2.1- IMPLEMENTATION METHOD

The system implementation followed the Client Centred Approach (CCA) method (Watson I. et al., 1992a, 1992b; Watson I. and Brandon P., 1993). According to Watson and Brandon (Watson I. and Brandon P., 1993), the CCA covers the full development life cycle of an KBS providing milestones to guide the project. These milestones refer to what the clients can see being demonstrated in terms of the project deliverables.

The CCA is a seven-stage method for developing KBSs. The seven stages of the CCA are named in terms of deliverables (what) that the client can expect throughout the project life. These stages are: i) the *holistic picture*; ii) the *skeleton system*; iii) the *demonstration system*; iv) the *working system*; v) the *usable system*; vi) the *saleable system*; and vii) the *embedded in use system.*

Following the CCA method, the implementation of Xtimela-CBR was a staged process measured by the deliverables produced at each stage. The CCA helped to keep the Client interested in the research during the implementation cycle and hence, the system acceptance.

The current implementation stage of Xtimela-CBR corresponds to the *working system* of the CCA method. This was the stage considered necessary for validating the research hypotheses.

7.2.2- DESCRIPTION OF Xtimela-CBR

7.2.2.1- Knowledge Structures

The system contains four kinds of knowledge structures, i.e.: *application cases*; *concepts*; *domain models*; and *procedures*. Application cases capture and represent relevant information from past applications. Concepts refer to the domain entities. Domain models capture the causal relationships among concepts. Procedures refer to the operators and the control of processing. To represent these knowledge structures in Xtimela-CBR, use was made of objects, object hierarchies, methods and functions which are provided by the Kappa-PC tool.

7.2.2.2- Application Case Database

The case library:
The case library consists of a repository of application cases. These application cases are represented and organised into a multilevel *kind-of* object hierarchy. This object hierarchy defines *classes* and *instances* at the bottom. *Classes* partly index applications cases in the case library. *Instances* store application cases represented as objects. *Inheritance* links classes and classes to instances.

At the skeleton stage of the system, the case library consisted of 60 real and fairly representative application cases from three sources: i) applications for renovation grants; ii) applications for disabled facilities grants; and iii) applications for minor works assistance. The decision to use at early development stage a case library containing 60 seed application cases was based on the following reasons:

- Some of the most representative systems using CBR have started running with small case libraries. For example, CLAVIER (Hennessy D. and Hinkle D., 1991,1992) started working with a case library containing of 20 autoclave load cases; ARCHIE's case library contained only 20 design cases (Pearce M. et al., 1992); and KRITIK (Goel A., 1989) started working with a case library containing 6 design cases.
- The need to cover a broad range of problems within the HRGS domain at an early implementation stage.
- To allow the system to acquire automatically additional application cases.
- To allow the validation of each application case added to the case library as consequence of the case acquisition process.

- To allow the Client to see what the system could provide and, hence, to keep its interest in continuing to support the research project.

The 60 seed application cases were selected from a sample of over 300 past applications successful processed by the Client. For example the 141 cases held by CASCADE (Simoudis E. and Miller J., 1991; Simoudis E., 1992) system were selected from a sample of 200 cases.

A simple selection criteria was adopted to select the seed cases so that they would be a representative and well-distributed set of past applications from that sample (Kriegsman M. and Barletta R., 1993). Thus, the selection of the 60 seed cases was based on the following criteria:

- *Location of the property*, by housing areas.
- *Type of grant.*
- *Fitness of the property* regarding the fitness standard for human habitation.
- *Cost of works*, by ranges of: 0 to £5,000.00; £5,001.00 to £10,000.00; £10,001.00 to £15,000.00; £15,001.00 to £20,000.00; and more than £20,001.00.
- *Decision on grant.*

Following the CCA method, the Xtimela-CBR's case library rose from 60 seed cases to a total of 155 application cases. These 155 application cases reasonably cover the major types of problems from three different sources, such as:

- 127 cases from applications for renovation grants;
- 18 cases from applications for disabled facilities grants; and
- 10 case from applications for minor works assistance.

Procedures to access the case library:
The procedures to access the case library are represented as functions using KAL language. They allow to create and store new application cases into the case library.

7.2.2.3- Application Case Representation

Application cases in the Xtimela-CBR describe specific knowledge from past grant applications. In choosing and implementing the Xtimela-CBR application cases, four questions had to be answered during the case acquisition and system's implementation:

- Firstly, can the past applications held by the Client be modelled in terms of application cases in a computer programme?

- Secondly, what component parts does an application case need to have and what kinds of information need to be encoded in those components. (Kolodner J., 1993)?
- Thirdly, what type of application cases are needed by the CBR method as described at the task structure?
- Finally, what representational structures are most useful in representing the different types of application cases (Kolodner J., 1993)?

The study carried out on the past application files held by the Client has shown that past applications form the main body specific knowledge within the HRGS domain. Each grant application represents a specific problem in terms of applicant(s), other occupants, property and location of the action. Therefore, each past application file: i) has a specific format to organise the information it contains; ii) presents a detailed description of the application, i.e., of the problem to be solve; iii) records the solutions found by Client's expert for that application; iv) records the outcomes of the application as consequence of its implementation; and v) captures how a Client's expert applied the rules laid down by the legal framework in conjunction with its skills in a specific situation.

The interviews carried out with Client's experts during the knowledge acquisition have confirmed what the postal survey indicated (chapter 2) about the use of past experience (solutions) held by past applications to assist the assessment of new grant applications. Client's experts often use past solutions to support the assessment of new grant applications. For example, the schedules of work for a new applications are derived from solutions used in past grant applications in similar situations. Therefore, Client's experts were more motivated to provide their knowledge in terms of specific experiences recorded in a grant application file. The study carried out on the past applications together with the interviews with experts confirmed what the postal survey indicated (chapter 2) about the similarities existing among grant applications within the same grant.

On the basis of the above discussion the first and second questions were answered. Thus, it was assumed that past grant applications held by the Client could be modelled in the Xtimela-CBR system as applications cases to support problem solving of new cases.

Regarding the second question, it was assumed that an application case in case library should encode the relevant information from a past grant application needed to perform the system's tasks. Taking into consideration the above discussion, the structure and content of an application case should be specified by:

- the structural organisation and content of the grant application files; and

- the range of tasks supported by the application cases in the system as described by the task structure.

Thus, application cases in the system's case library were structured into three main components:

1. **Description of the application** containing relevant information about: i) the property and works to be carried out; ii) the applicant's interest in the property and how the property is occupied; and iii) financial information of the occupants;
2. **Solution plan for the application:** containing information about: i) reasons for unfitness; ii) the grounds of unfitness; iii) the schedule of works; iv) means test; v) two estimates of the cost of works; vi) approved works including approved costs; vii) the amount of grant and amount not granted; viii) notice of provisional grant approval; ix) certificate of future occupation; x) notice of final grant approval; xi) inspection of the works reports; xii) certificate of completion of works; and xiii) payment notice.
3. **Outcomes of the application** containing information about: i) unforeseen works; ii) unforeseen costs; iii) confirmation that the standard of fitness was achieved; and iv) confirmation of completion of work according to the specification.

The three versions of the task structure presented in chapter 4 describes: i) how the assessment of grant applications tasks are performed; ii) what knowledge from application cases is required by each problem solving method; and iii) the range of tasks which the application cases are used for. Thus, taking into account the description provided by the task structure and the types of grants specified by the legal framework, it was found that eight types of application cases are required for the system's case library. They are: mandatory renovation application cases; discretionary renovation application cases; mandatory disabled facilities application cases; discretionary disabled facilities application cases; minor works thermal insulation application cases; minor works clearance application cases; minor works elderly application cases; and minor works repair and improvement application cases. These application case types come from three categories of grants corresponding to the scope of this research. Figure 7.2 shows the application cases required for the system's case library.

A final representation suitable for the three categories of application cases was found as a consequence of the case acquisition process described in chapter 5 and in taking into account: i) the components parts found for the application cases; and ii) the type of tasks supported by the application case types shown in figure 7.2. These three kinds of representations encode respectively:

- a total of 83 features with respect to application cases for renovation grants;
- a total of 54 features with respect to application cases for disabled facilities grants; and

- a total of 56 features with respect to application cases for minor works assistance.

These features are concepts, text, integers or real numbers.

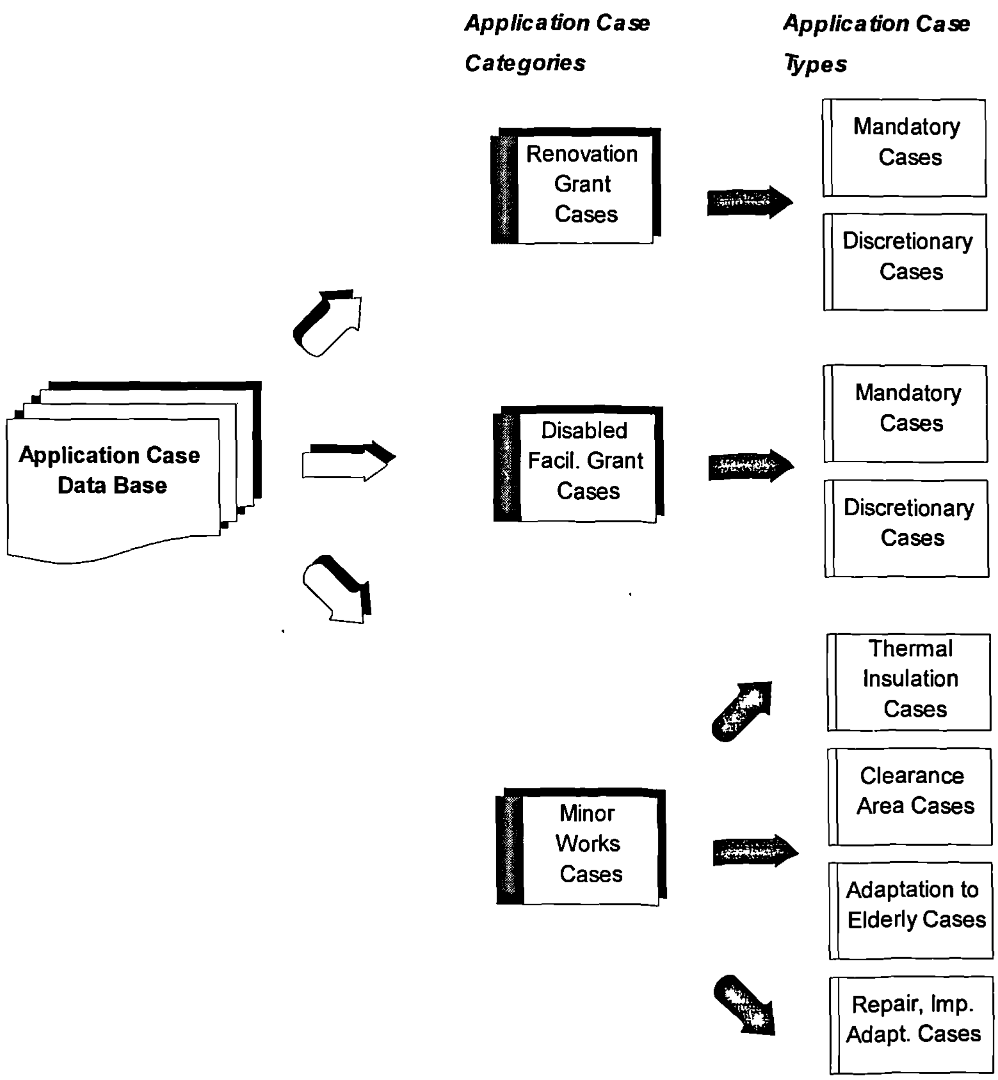


Figure 7.2: Types of application cases required by Xtimela-CBR system

Table 7.1 shows some of the features encoded in each category of the application cases.

Table 7.1: Some of the case representation features encoded in the system

Case Components	Feature	Renovation Grant	Disabled Facilities Grant	Minor Works Assistance
Description of the application	Application code	*****	*****	*****
	Applicants' name	*****	*****	*****
	Address of property	*****	*****	*****
	Certificate of future occupation	*****	*****	*****
	Type of grant sought	*****	*****	*****
	Construction era of the property	*****	*****	*****
	Interest in the land	*****	*****	*****
	Applicant type	*****	*****	*****
	Planning permission	*****	*****	*****

Case Components	Feature	Renovation Grant	Disabled Facilities Grant	Minor Works Assistance
Description of the application	Building Regulations approval.	*****	*****	*****
	Purpose of the application	*****	*****	*****
	Type of property	*****	*****	*****
	Special needs of the occupants	*****	*****	*****
	Housing needs for the area	*****	*****	*****
	Local circumstances	*****	*****	*****
	Future area action	*****	*****	*****
	Demolition order.	*****	*****	*****
	Grant history	*****	*****	*****
	Time elapsed from previous assistance	*****	*****	*****
	Type of use of the property	*****	*****	*****
	Previous contribution to grant	*****	*****	
	State of works at application date	*****		
	Property's age	*****		
	If included in a group repair	*****		
	Registration as disabled person		*****	
	Purpose of works			*****
	Applicants' age			*****
	Benefits in receipt by applicant(s)			*****
	Elderly person's age			*****
	Result of enquiry assessment	*****	*****	*****
	Fitness of the property	*****	*****	
Solution	Reasons for unfitness	*****		
	Grounds of unfitness	*****		
	Severity of unfitness	*****		
	Schedules of work Approved cost of works	***** *****	***** *****	***** *****
	Selected estimate	*****	*****	
	Applicable amount	*****	*****	
	Eligible income	*****	*****	
	Level of income	*****	*****	
	House conditions	*****		
	Eligibility of application	*****	*****	*****
	Reduction of grant	*****	*****	
	Amount of grant	*****	*****	*****
	Grant conditions	*****	*****	*****
	Works conditions	*****	*****	*****
	Grant decision	*****	*****	*****
	Amount not granted	*****	*****	*****
	Welfare recommendations		*****	
	Assessment of the adaptations		*****	
	Type of minor assistance			*****
	Certificate of future occupation	*****	*****	*****
	Condition of implementation			
Outcomes	Certification of completion	*****	*****	*****
	Standard of works	*****	*****	*****
	Unforeseen works	*****	*****	*****
	Economic results	*****		

*****= applicable blank=not applicable

The review of systems presented in chapter 3 identified several computer formalisms which have been used for representing cases. They include: feature:value representations like frames and objects (CADSYN, CHIRON, CHEF, HYPO, MEDIATOR and CASEY); form-like representations defined by set of fields and their values (ARCHIE and CLAVIER); and textual annotations. The feature:value representation in the form of objects was selected for representing application cases in the case library. The reasons for choosing this computer formalism were as follows:

- there are a number of systems using feature:value representations;
- the feature:value representation provides an easier way for representing and maintaining application cases;
- the content which should be encoded in each application case it is well known;
- the feature:value representation allows the developer to take advantage of the best features of the object-oriented programming paradigm, such as: inheritance features and incorporation of methods within the objects; and
- the desired implementation stage of the system corresponds to the experimental level.

Figure 7.3 shows a partial view of an application case encoded in the Kappa-PC object-like structure. Application case features are encoded into the Kappa-PC objects as slots. These slots store values acquired during the case acquisition process.

All of the conclusions in this section are drawn from work carried out on the sample of over 300 past grant applications which have been analysed.

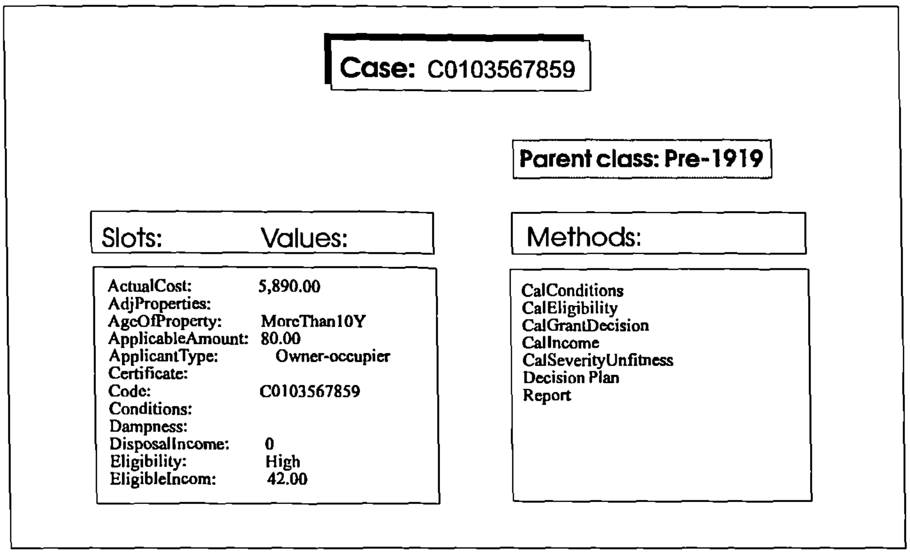


Figure 7.3: A partial view of the an application case encoded as Kappa-PC object

7.2.2.4- Application Cases Indexes

Assigning indices for the three categories of application cases was a lengthy task in the implementation process. Indices are data structures which provide a direct mapping from each specific feature of a new case to the stored cases having that feature. Retrieval of similar cases relies upon those index features (Zito-Wolf R. and Alterman R., 1993). Indices in Xtimela-CBR application cases come from the case representation features and play two main roles:

- organise the application cases in the case library; and
- direct the search and matching in the case library during the retrieval.

The selection of indices was carried out following the case acquisition described in chapter 5 and using simple criteria. This criteria included:

- ♦ The assignment of indices is determined by the nature of the current task being performed (Hammond K., 1989).
- ♦ The assignment of indices for the task of finding the schedules of work is based on the physical and functional attributes of typical a dwelling-house. Goel and Chandrasekaran (Goel A. and Chandrasekaran B., 1989) claim that in dealing with physical devices, the cases are retrieved not only by the physical attributes of the device, but also by its functional behaviour.
- ♦ Indices should be predictive of the solution that is being sought (Kolodner J., 1993).
- ♦ Indices should be concrete enough to be recognisable for retrieval purposes (Kolodner J., 1993).
- ♦ Indices should be abstract enough to make a case useful in a variety of situations for the current problem (Kolodner J., 1993).

Knowing where the indices come from (case representation features) and following the criteria above two sets of indices were selected for each of the tasks supported by application cases and for each of the three case categories. They are listed in tables 7.2 and 7.3. A description of these indices is presented as follows:

Indices with respect to the task of finding the schedules of work for renovation grants:

- Fitness of the property: refers to the assessment of the fitness of the property regarding the fitness standard.
- Construction era: refers to the era which the property was built.
- Reasons for unfitness: refers to the requirements which the property failed to comply with.
- Severity of unfitness: refers to the number of unfit requirements.

- Grounds of unfitness: refers the grounds that make a building component unfit.
- Building component: refers to the building component deemed unfit.
- Location of building component: refers to the location of the failure.

Indices with respect to the task of finding the schedules of work for disabled facilities grants:

- Reasonable and practicable: refers to assessment of the adaptations.
- Recommendation of the welfare services: refers to the adaptation recommended by the welfare services.
- Building component: refers to the building component affected by the adaptation.

Indices with respect to the task of finding the schedules of work for minor works assistance:

- Type of assistance: refers to the assistance sought by the application.
- Building component: refers to the defective building component.
- Location of building component: refers to the location of the failure.

Table 7.2: Indexing scheme for the task of finding the schedules of work

Indexing Scheme For Schedules of Work Task		
Renovation Grant Application Cases	**Disabled Facilities Application Cases**	**Minor Works Assis. Application Cases**
Fitness of the property	Reasonable and practicable	Type of assistance
Construction era	Recommendation of the welfare services	Building component
Reasons for unfitness	Building component	Location of building component
Severity of unfitness,		
Grounds of unfitness,		
Building component		
Location building component		

Indices with respect to the task of finding the solution of an application for renovation grant:

- Type of grant: refers to the grant sought by the application.
- Construction era: refers to the era which the property was built.
- Fitness of the property: refers to the assessment of the fitness of the property regarding the fitness standard.
- Degree of eligibility: refers to the merits of the current application.
- Future area action: refers to the action planned for the are where the property is located.
- Means test: refers to the test of resources of the relevant persons.

- Local circumstances: refers to the local circumstances associated with the current application.
- Adjoin properties: refers to the relation of the current property with adjoin properties.

Indices with respect to the task of finding the solution of an application for a disabled facilities grant:

- Type of grant: refers to the grant sought by the application.
- Fitness of the property: refers to the assessment of the fitness of the property regarding the fitness standard.
- Degree of eligibility: refers to the merits of the current application.
- Future area action: refers to the action planned for the are where the property is located.
- Means test: refers to the test of resources of the relevant persons.
- Needs of disabled person: refers to the specific needs of the disabled occupant.

Table 7.3: Indexing scheme for the task of finding the solution for the current application

Indexes For The Three Classes Of Application Cases		
Renovation Grant Application Cases	**Disabled Facilities Application Cases**	**Minor Works Assis. Application Cases**
Type of grant	Type of grant	Type of grant
Construction era	Fitness of the property	Type of assistance
Fitness of the property	Degree of eligibility	Degree of eligibility
Degree of eligibility	Future area action	Applicant' age
Future area action	Means test	
Means test	Needs of disabled person	
Local circumstances		
Adjoin properties		

Indices with respect to the task of finding the solution of an application for minor works assistance:

- Type of grant: refers to the grant sought by the application.
- Type of assistance: refers to the assistance sought by the application.
- Degree of eligibility: refers to the merits of the current application.
- Applicants' age: refers to the age of the applicant or the beneficiary of the grant.

7.2.2.5- Case Library Organisation

This section discusses how the application cases were organised into the case library in order to make efficient the search and matching of a similar case. Goel A. (Goel A., 1989) in discussing the KRITIK system stated that for a large memory, design cases need to be organised into hierarchies. The organisation of design cases in hierarchies allowed to better discriminate them in KRITIK system. Simoudis and Miller (Simoudis E. and Miller J., 1991) for the STAIN CBR system organised cases into a hierarchy of clusters. These authors claim that hierarchical organisations have two advantages: i) they partially index cases in the system; and ii) they make case retrieval more efficient. A number of other applications follow the organisation of cases into case hierarchies. These applications include: CADSYN (Maher M. and Balachandran B., 1994); ARCHIE (Pearce M. et al., 1992); CASCADE (Simoudis E., 1992); and HYPO (Ashley K. and Aleven V., 1992). All authors of these systems claim that a hierarchical organisation:

- is required when the case library reaches an appreciable size; and
- allows the implementation of serial search procedures for retrieval purposes.

Taking into account the experience of systems mentioned above and the guidelines described in chapter 3 the system's case library was organised into a multilevel hierarchy.

Figures 7.4 and 7.5 show two partial views of this hierarchical organisation implemented using Kappa-PC tool. Thus, case library organisation is described as follows:

1. **Top level:** this is the most general level which defines the general concept of an application case.
2. **Second level:** corresponding with the categories of application cases found in the domain. Thus, three classes were defined at this level.
3. **Third level:** corresponding with the different processing stages followed by the Client in assessing an application. Thus, five classes were defined at this level: i) final payment; ii) interim payment; iii) final approval; iv) provisional approval; and v) in processing.
4. **Fourth level:** corresponding with the construction era which the property belongs. Thus, two classes were defined at this level: i) houses built before 1919; and ii) houses built after 1919.

These levels partially index application cases and direct the search and matching in the case library. The organisation of the case library in partitions is so that only the relevant portion of it is accessed by the retrieval procedures.

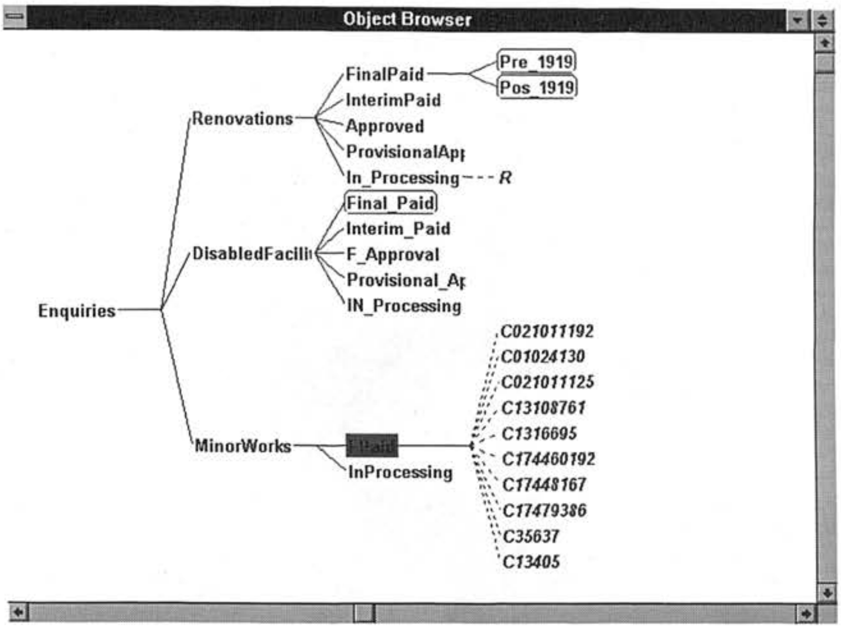


Figure 7.4: Partial views of the case library hierarchical organisation implemented in Kappa-PC shell

As illustrated by figures 7.4 and 7.5, the hierarchical organisation adopted for the case library provides a visualisation of the relationships between application cases which share some features in common. Thus, it is possible to see which cases are more closely related to each other.

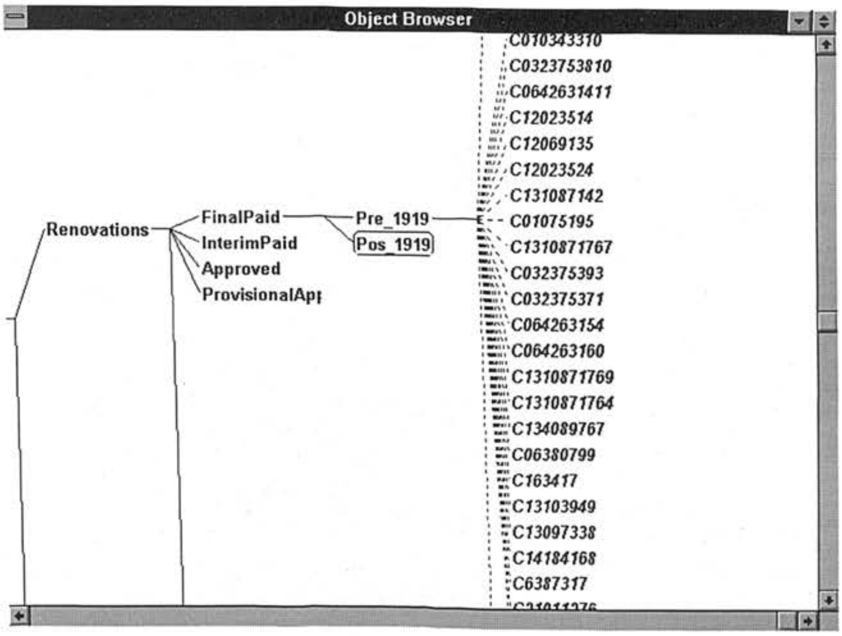


Figure 7.5: Partial views of the case library hierarchical organisation implemented in Kappa-PC shell

7.2.2.6- Case-Base Reasoner

The case-base reasoner has knowledge about how to retrieve a similar application case, adapt solutions stored in an application case, present and store an application case once it has been successfully processed by the system. A description of the implementation of the components of the case-base reasoner is provided in the following sections.

7.2.2.6.1- Case-Base Retriever

The main issue of a case-base retriever is how efficiently it can retrieve only those cases which are useful to the current task (Goel A. et al., 1991). The task structure presented case retrieval as a search problem, i.e., the system should know how to search the case library to find application cases with the potential to match the current situation. Kolodner (Kolodner J., 1993) pointed out that the retrieval of cases from a case library can be seen as a search problem. She added, that case retrieval requires a combination of three main steps: i) *search in the case memory* to find potentially matching cases; ii) *matching relevant cases*; and iii) *ranking the retrieved cases*. At the present implementation stage of Xtimela-CBR, the case-base retriever does not perform the third step defined above, because it just retrieves the most *similar* application case. A similar case in Xtimela-CBR is one that is similar to the current problem in its relevant index features. Taking into account the description provided by the task structure, Xtimela-CBR's case-base retriever performs the retrieval task in three main steps:

1. it searches the case library for a potentially similar application case;
2. it matches the most similar application case similar case; and if successful
3. it retrieves the solution from the selected case.

Figure 7.6 shows how these steps were implemented in the system. Xtimela-CBR case-base retriever includes six different retrieval procedures. These procedures were developed for the range of tasks which are supported by the application cases. These retrieval procedures are associated with the indexes presented in section 7.2.2.4 and with the organisation of the case library presented in section 7.2.2.5. They integrate searching and matching as it is described as follows:

1. Searching the case library:
It searches the case library to select a cluster of past application cases which share general common features. For example, in the context of the task of finding the schedules of work, this search is directed by: i) *the type of grant* sought by the application; and ii) *the construction era* of the property.

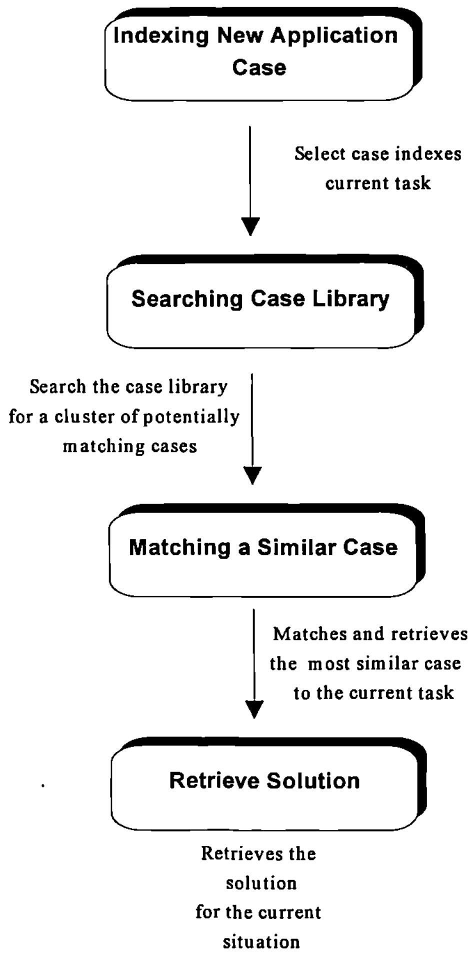


Figure 7.6: Procedure followed to implement the case-base retriever

2. Searching and matching in the selected portion of the case library:

It searches the selected cluster for a potentially matching application case in the context of the problem being solved. During the search in the selected cluster, each potential matching application case is judged by its similarity to the current situation by comparing each of the corresponding relevant feature:value(s). Hence, the retrieval procedure, assesses the similarity between a potential matching application case and the current situation based on a task-specific criteria. Leak (Leak D., 1991) pointed out that CBR systems require criteria for deciding the similarity of cases and there is a wide agreement among researchers that these similarity criteria can be task-specific.

The criteria used to assess the similarity of application cases is based on both:

- the importance of each index feature selected in the context of the current task; and
- the degree of match of each of the corresponding features of the new situation and the old application case for the current task goal.

In order to measure the degree of match, the case based retriever uses a direct similarity metric inspired by the Prodigy system (Veloso M. and Carbonell J., 1991). This direct similarity metric is described as follows:

A past application case (C_p) directly matches a new situation in the context of the current case (C_n) *if*:

- the relevant indexes for the current task of C_n $\{f_1, f_2,.....,f_n\}$ are the same as its corresponding features in the C_p $\{f'_1, f'_2,.....,f'_n\}$.
- each relevant feature:value of C_n $\{f_i:v_i\}$ fully matches each relevant feature:value of C_p $\{f'_i:v'_i\}$.
- in aggregate, all of the relevant feature-value pairs of C_n $\{f_1:v_i, f_2:v_j,.....,f_n:v_k\}$ matches the relevant feature:value of C_p $\{f'_1: v'_i, f'_2:v'_j,.....,f'_n:v'_k\}$.

For example, in the context of the task of finding the schedules of work the relevant features being used for similarity judgement are: i) the *ground of unfitness*; ii) the *type of building component* deemed defective; iii) the *failure type*; and iv) the *location* of the defective building component.

3. Retrieval of solution:

According to the measure of similarity achieved in step 2, an application case is only retrieved where the degree of match of each of the corresponding index feature of the new situation and of the retrieved application cases scores 100%, either individually or on aggregate. It is expected that the retrieved case contains the solution sought. For example, in the schedules of work task it is expected that the retrieved application case contains a schedule of works which satisfies the specification of the current problem: i) *ground of unfitness*; ii) *defective building component*; iii) *failure*; and iv) *location.*

A partial view of how the system derives the schedules of work for a given dwelling-house deemed unfit for human habitation is presented below.

Xtimela-CBR activates the case-base retriever

```
.............................................................
ClearTranscriptImage(Transcript6);
PostBusy(ON ,"Calculating The Schedule Of Renovation Works");
Wait(1);
PostBusy(OFF);
ClearList(Works:Comps);
GetInstanceList(BuildComps,Works:Comps);
SendMessage(BuildComps,CalRenovWorks);
ClearTranscriptImage(Transcript4);
ClearTranscriptImage(Transcript7);
SetValue(BuildComps:Schedule,RenovationWorks:Repair,RenovationWorks:Stable,
RenovationWorks:Dampness,RenovationWorks:Lighting,RenovationWorks:Ventilation,
RenovationWorks:Heating,RenovationWorks:Cooking,RenovationWorks:Water,
RenovationWorks:WC,RenovationWorks:Bath,RenovationWorks:Drainage);
.............................................................
.............................................................
```

The case-base retriever searches the case library for a partition of candidate cases, matches a similar application case and retrieves a schedule of work

```
.............................................................
ClearList (Global:Reason);
 ResetValue(R:ConstructionEra);
 SetValue(Global:Reason , R:ReasonOfUnfitness);
  R:ConstructionEra = PostMenu (" Select the construction era which
  the property belong to" , Pre_1919 , Post_1919, Unknown);
 EnumList(Global:Reason,reason,
 {
   If reason #= SeriousDisrepair
    Then SendMessage (RenovationWorks,CalRepair);
     If reason #= StructuralUnstable
   Then SendMessage (RenovationWorks,CalStable);
   If reason #= PrejudicialDampness
.............................................................

.............................................................
  ClearList(RenovationWorks:Stable);
  ClearList(Global:SelectedCases);
  EnumList(R:Stable , stable ,
   {
    Works:Length1 = StringLength(stable );
    Let[elem SubString(stable,1,3)]
       [fail SubString(stable ,4,Works:Length1)]
       {
        Works:Length2 = StringLength(fail);
        Works:Elem = elem;
        Works:Fail = fail;
       };
    If  R:ConstructionEra #= Pre_1919
       Then
        {
         PostBusy(ON,"Consulting The Case Database Of Past Applications To Retrieve The Renovation Works");
         Wait(1);
         PostBusy(OFF);
         GetInstanceList( Pre_1919 ,Global:SelectedCases);
         Works:Case = SelectList(Global:SelectedCases, case,
           Member? (case:Stable , stable) );
         If Not(Null?(Works:Case))
          Then
           {
           Let [ cas Works:Case ]
            {
             EnumList( cas:RStable ,renov ,
              {
               Works:Length3 = StringLength(renov );
               Let[eleme SubString(renov ,(Works:Length3 - 2),Works:Length3)]
                  [ren SubString(renov ,1,(Works:Length3 - 3))]
                {
                 If  eleme #= Works:Elem And
                   FindSubString(renov,Works:Fail,1)> 0 And
                   Not(Member?(RenovationWorks:Stable,renov))
                   Then
                    {
                     Works:Renovation = SelectList(Works:Comps ,comp,
                       comp:Code#=Works:Elem);
                     Let[comp Works:Renovation]
```

```
            {
            ClearTranscriptImage(Transcript4);
            ClearTranscriptImage(Transcript7);
            DisplayText(Transcript4,FormatValue("%s",comp:ComponentName));
            DisplayText(Transcript7,FormatValue("%s",Works:Fail));
            AppendToList (RenovationWorks:Stable,
               ren #comp:ComponentName);
            AppendToList(R:RStable,renov);
            };
          };
        };
      });
    };
    }
  Else SendMessage(BuildComps ,CalS);
 };
If R:ConstructionEra #= Pos_1919
  Then
   {
   PostBusy(ON,"Consulting The Case Database Of Past Applications To Retrieve The Renovation Works");
   Wait(1);
   PostBusy(OFF);
   GetInstanceList( Pos_1919 ,Global:SelectedCases);
.............................................................
.............................................................
```

The above procedure shows that when the case-base retriever fails to retrieve a similar application case then the case-base reasoner activates the case-base adapter through the function *"SendMessage(BuildComps, CalS)"*. Xtimela-CBR uses the output of the assessment of the fitness task as a probe into the case library to search for application cases that match the current problem.

Figure 7.7 shows a screen image of a set of retrieved schedules of work for a specific property deemed unfit.

7.2.2.6.2- Case-base Adapter

The task structure and the task knowledge discussed respectively in chapters 4 and 5 presented the case adaptation for the schedules of work task as a substitution problem. Kolodner (Kolodner J., 1993) defined the substitution methods as:

> *"The process of selecting and implementing a replacement for some part of an old solution".*

Some of the systems discussed in chapter 3 use substitution methods for carrying out the adaptation. These applications include: CHEF (Hammond K., 1989); CLAVIER (Hennesey D. and Hinkle D., 1991, 1992); CASEY (Koton P., 1988); PERSUADER, (Sycara K., 1987); and JUDGE, (Bain W., 1989). These systems have shown that there are several kinds of possible substitutions. They included: i) a component of an old solution which can be substituted; ii) the amount of some component which can be substituted; or iii) a whole group of components or amounts which can be substituted. To carry out these substitutions a number of methods have been employed. They include: i) the CHEF system uses *reinstantiation* as a substitution method to adapt an old recipe; ii) the CLAVIER system uses *case-based substitution*, i.e., other cases to suggest substitutions; iii) JUDGE system uses *parameter adjustment* to adapt an old sentence. Oxman (Oxman R.E., 1991) proposed three adaptation methods for housing design, such as: i) *parametric modification*, which changes values of design variables; ii) *substitution adaptation*, which replace elements of a design solution; and iii) *topological adaptation.*

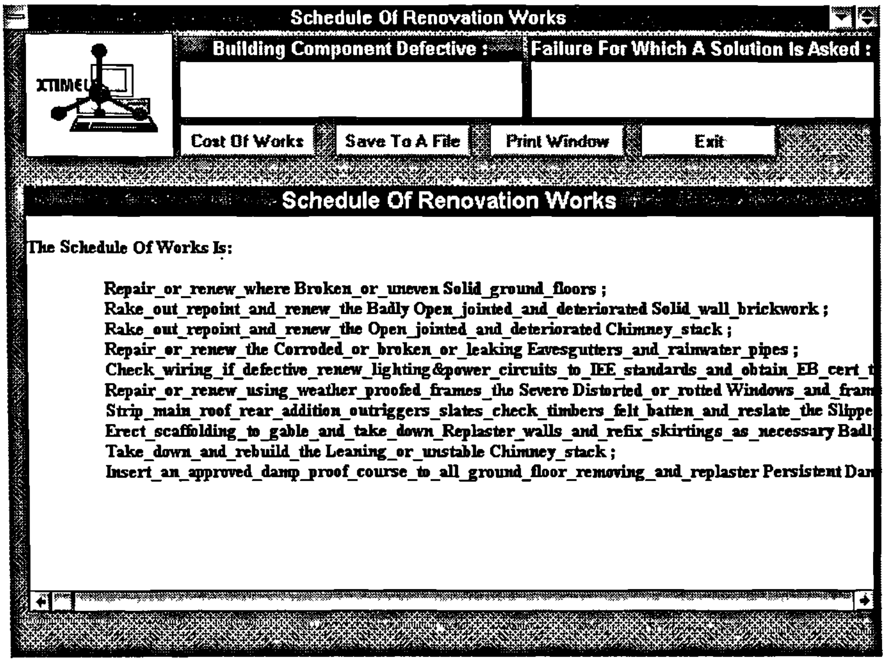


Figure 7.7: A screen of Xtimela-CBR showing set of retrieved schedules of work

To implement the adaptation of schedules of works the *case-based substitution* method was selected. The reasons for choosing it are:

- The description presented in chapter 5 shows that: i) schedules of work found in past application can be decomposed in several primitive components; and ii) by modifying the type of work component different specifications of schedules of work can be obtained for the same ground of unfitness. Thus, adaptation of a schedule of work can be carried out by substituting the type of work component.
- Because there are similarities between schedules of work found in the past applications within the same reason for unfitness, application cases stored in the case library can be used to suggest types of works substitutions.
- As illustrated by CLAVIER system, case-based substitution method uses other useful cases to suggest types of work load substitutions.

Case-based substitution method in Xtimela-CBR performs the adaptation by substituting the *type of work* component of an old solution for another which can be selected from a list of types of work provided by the retrieved *useful* past application cases. Here, a past application case is considered *useful* if is similar to the new problem in some of the relevant indexes selected for the current task. Figure 7.8 illustrates how the case-base adapter was implemented in the system.

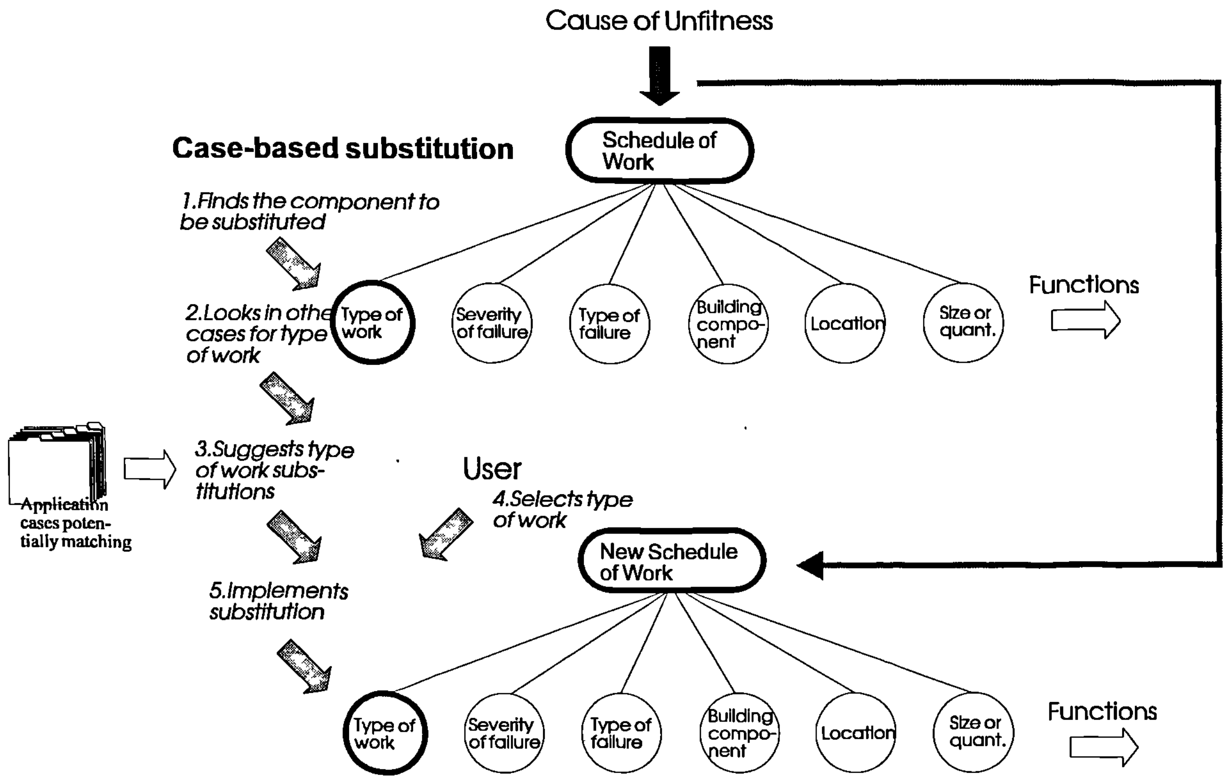


Figure 7.8: How the case-base adapter was implemented in the system

As shown in figure 7.8, the case-base adapter finds the component to be replaced and looks for types of work in the potentially matching application cases containing schedules of work for the current building component and reason for unfitness. Then, it suggests a set of substitutions to the user. The user selects a type of work and the system implements the substitution. Xtimela-CBR case-base adapter incorporates three different adaptation procedures corresponding to the application case categories.

7.2.2.6.3- Case-Base Reporter

The case-base reporter presents to the user the relevant information of the solution found for the current application as specified in chapter 5. It uses three different procedures to present information corresponding to each category of application cases. These procedures display the relevant information contained in the solution plan in an appropriate format. To access the other features of the current application case, the user has to store the application case and then to access it through the case-base manager. In fact, the application case stores a large amount of processed information which the user might be interested in and which is not presented by the case-base reporter. Figure 7.9 shows a partial view of a screen image of a solution plan presented by the case reporter.

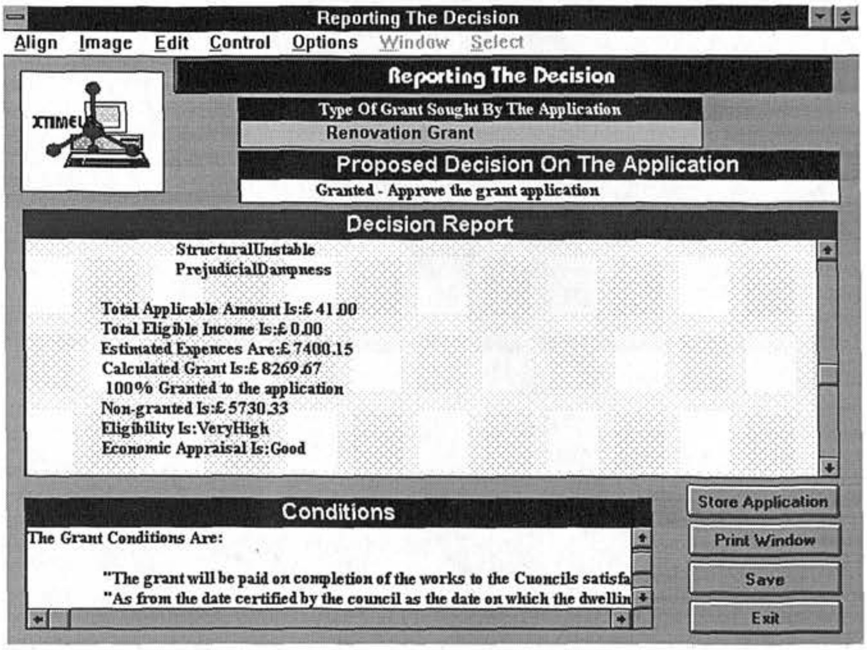


Figure 7.9: Partial view of a screen image of a solution presented by the case reporter

7.2.2.7- Case-Base Manager

The case base manager has an user interface that provides access to its components. A screen of this interface is shown by Figure 7.10. Components of the case-based manager are described in following subsections.

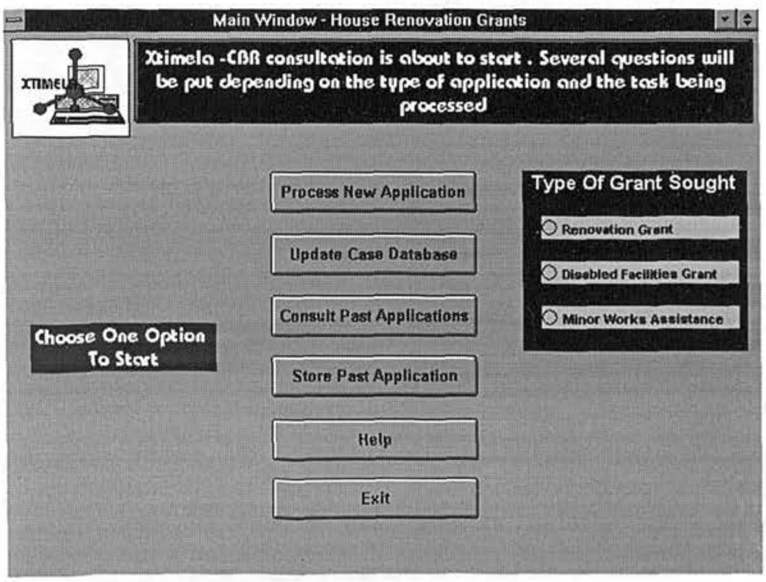


Figure 7.10: A screen image of the interface to access the case-base manager

7.2.2.7.1- Application Case Generator

The case generator uses the access procedures to generate and store a new application case in the case library. The new application case is stored temporarily in an appropriate partition of the case library so that it can be computed by the system's components. A new application case is stored in the "*In_Processing*" partition of the case library, with an appropriate case representation structure inherited from the corresponding application case category. A partial view of one of the access procedures used by Xtimela-CBR to create a new case is presented below.

The case generator starts by asking the general data about the enquiry

```
.........................................................
ResetValue(Global:Address);
 ResetImage(RadioButtonGroup4);
 PostInputForm("Enter The Application's General Information:",
    Global:EnquiryCode,"Enquiry Code Number:",
    Global:Name, "Applicant's Name(Name,Surnane):",
    Global:Address,"Property Address:");
 Global:NewEnquiry = C#Global:EnquiryCode ;
 .........................................................
 .........................................................
```

The case generator create a new application case in the case library

```
.................................................................
{
   PostMultipleSelection( "Select The Grant Which Is Sought By The Applicant:",
      Global:GrantType, "Renovation Grant", "Disabled Facilities Grant",
      "Minor Works Assistance", CANCEL );
   If knownValue?( Global:NewEnquiry )
    Then {
        If ( Global:GrantType #= "Renovation Grant" )
         Then{
             MakeInstance( R, Renovations );
             SetValue(R:Name , Global:Name);
             SetValue(R:EnquiryCode = Global:EnquiryCode);
             SetValue(R:PropertyAddres = Global:Address);
             StartWindow3( );
             }
         Else PostMessage( "Can not create a new enquiry" );
        If ( Global:GrantType #= "Disabled Facilities Grant" )
          Then {
              MakeInstance( DF, DisabledFacilities );
              SetValue(DF:Name , Global:Name);
              SetValue(DF:EnquiryCode , Global:EnquiryCode);
              SetValue(DF:PropertyAddres , Global:Address);
              StartWindow4( );
              }
          Else PostMessage( "Can not create a new enquiry" );
         If ( Global:GrantType #= "Minor Works Assistance" )
          Then {
  .........................................................................
```

7.2.2.7.2- Case-Base Browser

The case-base browser component incorporates a set of procedures for consulting and updating the application cases stored in the case library. The case library in Xtimela-CBR can be updated in two ways: i) by the system, by adding a new application case in the library after each problem solving session; and ii) by the user through the case-base browser by changing the information contained in the past application cases.

7.2.2.8- Brief Description Of How Xtimela-CBR Solves Some Of The Tasks Of The Assessment Of Grant Applications

7.2.2.8.1- Enquiry Eligibility

Xtimela-CBR uses a set of procedures for evaluating the enquiry eligibility for a grant application. A partial view of how Xtimela-CBR evaluates an enquiry for a renovation grant is presented below. Firstly, the system prompts the user with a set of questions about the enquiry regarding both, the applicant and the property.

Xtimela-CBR begins by asking the enquiry data

```
.................................................................
ResetValue(R,DifPurpose);
  ResetValue(R,Purpose);
  AskValue(R ,Age);
  AskValue(R, Planning&Regulations );
  AskValue(R, Interest );
  AskValue( R, ApplicantType );
  AskValue( R, StateOfWorks );
  AskValue(R,Served189or190 );
  AskValue( R, Residence );
  AskValue(R, Demolition);
  AskValue(R,Defective);
  AskValue(R,GrantHistory);
  AskValue(R,Purpose);
  If R:GrantHistory #= Yes
   Then AskValue(R:DifPurpose);
  PostMessage ( "To proceed for the preliminar Assessment first press OK and then the Preliminar Assessment button.");
 };
.............................................................................;
.............................................................................
```

Then, the system evaluates the eligibility of an enquiry by searching through the grant conditions model and evaluating every condition required to make an enquiry eligible for a grant application.

Xtimela-CBR evaluates the enquiry data

```
..........................................................................
Session4:Title = "Reporting Preliminar Assessment";
 PostBusy(ON," Doing The Preliminar Assessment Of The Application");
 ResetValue(R:PreliminarAssessment);
 If R:Served189or190 #= Yes
  Then  SendMessage(Renova_Cond , CalNotice)
  Else
   {
   If R:GrantHistory #= Yes
    Then SendMessage(RenCond2 , AssessConditions)
    Else SendMessage(RenCond1 , AssessConditions);
   };
  PostBusy(OFF);
..........................................................................

..........................................................................
ResetValue(Global:NotSatisfied);
 ForAll [ cond | RenCond2]
  {
   ResetValue(cond:Decision);
   SendMessage(cond , CalCondition);
  };
  If AreAll?( [dec|RenCond2],dec:Decision #= Acceptable)
   Then
    {
     R:PreliminarAssessment = "Proceed With Application";
    DisplayText(Transcript1,FormatValue("\n\t%s%s\n" , "Application From :",R:Name));
    DisplayText(Transcript1,FormatValue("\t%s%s\n" , "Address of Property : ",  R:PropertyAddress));
..........................................................................
 If KnownValue?(Self:Condition)
  Then
   {
   EnumList(Self:RequiredSlots,slot,
    If Null?(R:slot)
     Then AskValue(R:slot));
   If Null?(SelectList(Self:RequiredSlots,slot,
      Null?(R:slot)))
    Then
     {
      If EvaluateKAL(Self:Condition#" ;" )
       Then Self:Decision = Acceptable
       Else Self:Decision = NotAcceptable ;
     };
    }
   Else Self:Decision = NotKnown;
  };
```

Figure 7.11 shows a partial view of the presentation of a conclusion about the eligibility of an enquiry for a renovation grant.

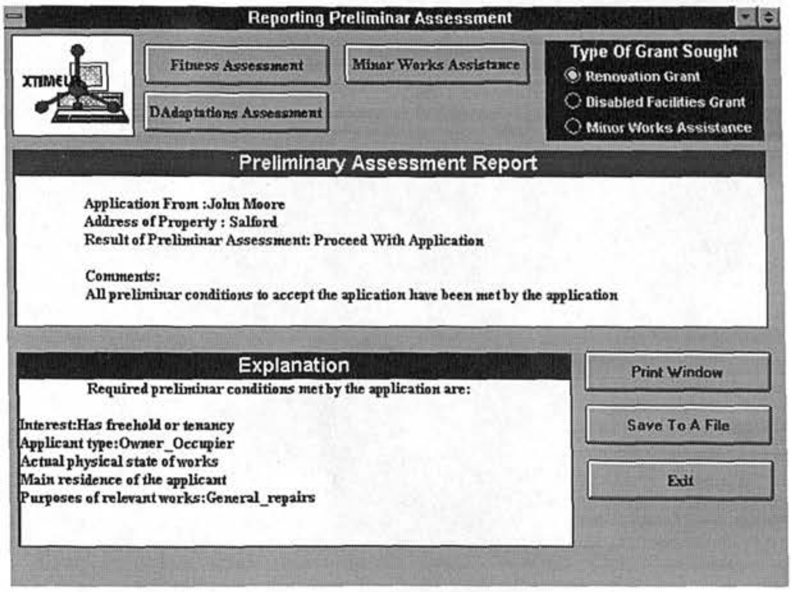


Figure 7.11: Partial view of a presentation of a conclusion about the eligibility of an enquiry for a renovation grant

As can be seen in figure 7.11, the system provides an explanation for the conclusion achieved as a result of the evaluation carried out.

7.2.2.8.2- Assessment of the Fitness of the Property

Xtimela-CBR incorporates a set of procedures which use the fitness standard model and the building component model to perform the assessment of the fitness of a dwelling-house. In assessing the fitness of the current dwelling-house, the system begins by prompting the user with a set of simple questions designed to find the evidence of failures. These questions are carried out in a similar manner as undertaken by human experts when inspecting a dwelling-house. This task was programmed to be carried out during the inspection of the house, so the user is just required to answer questions by choosing between "Yes" or "No" according to his or her observations. These questions were designed to guide the assessment of the condition of the dwelling-house with respect to the fitness standard. During the inspection the system computes the grounds of unfitness using the abductive assembly method and evaluates the fitness regarding each requirement of the fitness standard. The procedure below shows how the system implements these steps.

Xtimela-CBR begins the inspection on site, finds the evidence of failures, and computes the grounds of unfitness and the fitness with respect to each requirement

```
.........................................................................
ClearTranscriptImage(Transcript24);
 PostBusy(ON,"We Are Going To Assess The Fitness Of The Property According The Fitness Standard");
 Wait (3);
 PostBusy(OFF);
 PostBusy(ON,"Assessing Repair Requirements");
```

```
Wait(1);
PostBusy(OFF);
ForAll [ rep|Repairs ]
  SendMessage ( rep, Test);
PostBusy(ON,"Assessing Structural Stability Requirements");
Wait(1);
PostBusy(OFF);
ForAll [ stable|StructuralStability ]
  SendMessage ( stable, Test);
PostBusy(ON,"Assessing Dampness Requirements");
Wait(1);
PostBusy(OFF);
..........................................................................
..........................................................................
AskValue( Self:Structure);
 AskValue( Self:Structure1);
 AskValue( Self:Structure2);
 AskValue( Self:Fascia);
 If  Self:Structure #= Yes Or Self:Structure1 #= Yes Or
     Self:Structure2 #= Yes
    Then
     {
      Self:Condition = Unfit ;
      Self:Marginal= Unfit;
     }
    Else
     {
      Self:Condition = Fit  ;
      Self:Marginal = Fit
     };
 If Self:Structure #= Yes
    Then AppendToList (R:Repair,"G50 Badly Rotted_or_weakened");
 If Self:Fascia #= Yes
    Then AppendToList (R:Repair,"G58 Rotted");
  If Self:Structure1 #= Yes
    Then AppendToList (R:Repair,"G59 Rotted");
 If Self:Structure2 #= Yes
    Then AppendToList (R:Repair,"G61 Fractured");
};
..................................................................................
..................................................................................
 ResetValue(Self:ReasonOfUnfitness);
  ForAll[x|Self]
   If x:Marginal #= Marginal
    Then AppendToList(Global:Marginal,x);
  If AreAll?([ x|Self ],x:Condition #= Fit) And
     LengthList (Global:Marginal) = 0
    Then Self:Fitness = FitForHumanHabitation ;
  If  IsThereAny? ( [y |Self ] ,y:Condition #= Unfit) Or
      LengthList (Global:Marginal ) >= 5
   Then
      {
       Self:Fitness = UnfitForHumanHabitation ;
       Self:ReasonOfUnfitness = SeriousDisrepair;
       AppendToList(FitnessStandard:ReasonOfUnfitness,SeriousDisrepair);
      };
    If Null?(Self:Fitness) And Null?(Self:ReasonOfUnfitness) And
     LengthList (Global:Marginal) < 5 And LengthList (Global:Marginal) > 0
```

```
Then
 {
  Self:Fitness = FitForHumanHabitation;
  Self:ReasonOfUnfitness = SeriousDisrepair;
  AppendToList(FitnessStandard:ReasonOfUnfitness,SeriousDisrepair);
 };
.........................................................................................
```

In the next step the system computes the overall fitness of the dwelling-house and provides:

- a complete report on the fitness of the house; and
- the new application case with the values for the indexing scheme on which the case-base retrieval will focus during the schedules of work task.

A partial view of how the system computes the overall fitness is presented below.

```
.........................................................................................
ResetValue(FitnessStandard:Fitness);
 EnumSubClasses(FitnessStandard, FIT, SendMessage(FIT ,
        Test) );
 GetSubClassList(FitnessStandard,Global:Fitness);
 EnumList(Global:Fitness, x ,
  If x:Fitness #= UnfitForHumanHabitation
   Then  AppendToList( Global:Unfit , x )) ;
 If (LengthList(Global:Unfit) > 0)
   Then FitnessStandard:Fitness = UnfitForHumanHabitation
   Else FitnessStandard:Fitness = FitForHumanHabitation ;
 ......................................................................
```

The fitness assessment report includes information about: i) the dwelling-house; ii) the requirements of the fitness standard; iii) the conclusion about the fitness of the dwelling-house; and, if unfit, iv) the reasons for unfitness; and v) the grounds of unfitness. A partial view of this reports for a given application is shown in figure 7.12.

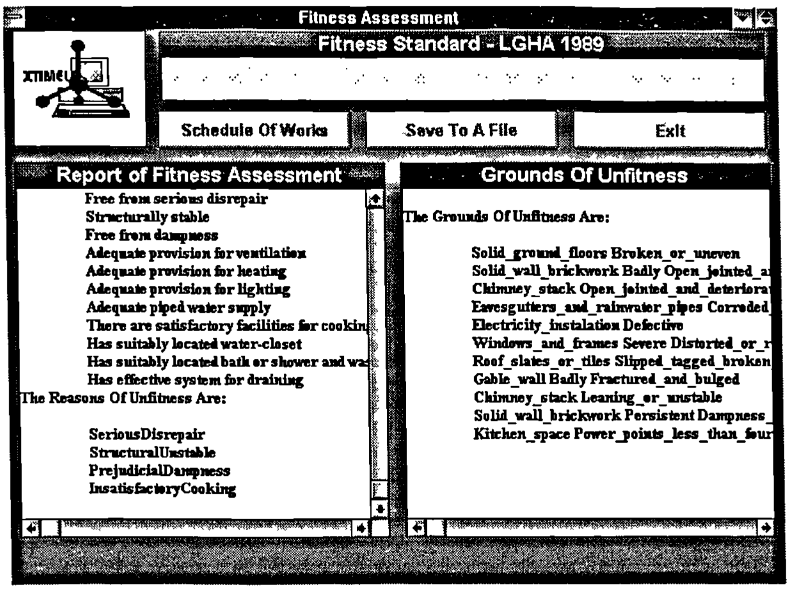


Figure 7.12: A partial view of the fitness assessment report provided by the system

Once the fitness assessment is completed the system's control asks the user to proceed to the schedules of work task. Then, the system control calls the case-base retriever to perform the schedules of work task as outlined in section 7.2.2.6.1. The schedules of work computed by the case base retriever should be priced by the two different contractors. Where the current application case is for a disabled facilities grant, the system, instead of computing the fitness assessment, computes the assessment of the required adaptations in order to find if:

- the adaptations are necessary and appropriate; and
- the adaptations are reasonable and practicable considering the condition and age of the dwelling-house.

7.2.2.8.3- Cost of Works

Xtimela-CBR computes the total amount of costs required to implement the schedules of work by using a set of procedures. It incorporates three different procedures corresponding to the categories of application cases. The costs are computed using the individual costs approved for each schedule of work, on the basis of the best contractor estimate and previous experience. The system prompts the user to allocate a cost to each schedule of work, and then computes the total cost including tax (VAT).

7.2.2.8.4- Economic Analysis

After computing the cost of works for renovation grant applications, the system proceeds by evaluating the economic merits of the renovation works against other available options such as: i) demolition; or ii) demolition with redevelopment; and iii) maintaining the condition of the house just with small scale repairs. Thus, the system incorporates a procedure that computes the NPVs for each option and finds a conclusion on the merits of the renovation by comparing the NPV of the renovation option with: i) the NPVs of alternative options; and ii) the limit of £20,000.00 defined by the legal framework. A partial view of how Xtimela-CBR implements the economic analysis is presented below.

Xtimela-CBR begins by computing the NPV for each option

```
..................................................................................................
ResetValue(EconomicAnalysis:Appraisal);
 Wait(1);
 SendMessage(EconomicAnalysis , EconomicAppraisal);
 Let[ anse PostMenu("Would You Like To Calculate The Value Of Grant",Yes,No,CANCEL)]
  If anse #= Yes
   Then PostMessage("First press OK and then button Means Test")
...............................................................................................................
...............................................................................................................
EconomicAnalysis:CostRen = BuildComps:ApprovedCost ;
 AskValue(EconomicAnalysis:IncMarketValue1);
 AskValue(EconomicAnalysis:RepairCosts15);
 AskValue(EconomicAnalysis:DiscountRate);
```

```
AskValue(EconomicAnalysis:ExpectedLife1);
 If EconomicAnalysis:ExpectedLife1 #= Yes
 Then
  {
  Let [ ben EconomicAnalysis:IncMarketValue1]
      [ ren EconomicAnalysis:CostRen]
      [ rate  EconomicAnalysis:DiscountRate]
      [ rep EconomicAnalysis:RepairCosts15]
   {
   If rate #= 10
    Then EconomicAnalysis:NPVRenovation =(ren - ben) + 0.26*( rep- 0) + 0.06*(ren - ben);
   If rate #= 8
    Then EconomicAnalysis:NPVRenovation =( ren - ben) + 0.34*(rep - ben) + 0.11*(ren - ben);
   };
  }
 Else EconomicAnalysis:NPVRenovation = Unknown;
 DisplayText(Transcript13,FormatValue("%8.2f",EconomicAnalysis:NPVRenovation));
 };
.........................................................................................
```

Then, Xtimela-CBR computes the economic merits of the renovation action

```
.........................................................................................
SendMessage(EconomicAnalysis, CalNPVMaintain);
 ResetImage(RadioButtonGroup1);
 AskValue (EconomicAnalysis:Unforseen);
 Let [ renov  EconomicAnalysis:NPVRenovation]
     [ maint  EconomicAnalysis:NPVMaintain ]
     [ demol  EconomicAnalysis:NPVDemolition]
     [ unf    EconomicAnalysis:Unforseen]
     [ cost   EconomicAnalysis:CostRen ]
  {
  DisplayText(Transcript16,FormatValue("%2.2f",EconomicAnalysis:DiscountRate));
  DisplayText(Transcript17,FormatValue("%8.2f",(cost+unf) ));
  If  renov < maint And renov < demol And
    (cost + unf) < 20000
   Then
    {
    EconomicAnalysis:Appraisal = VeryGood ;
    DisplayText(Transcript18,FormatValue("\n%s\n","The Decision on economic analysis was based on:"));
    DisplayText(Transcript18,FormatValue("\n\t%s","NPV for renovation less than NPV for demolition"));
    DisplayText(Transcript18,FormatValue("\n\t%s","NPV for renovation less than NPV for maintaining"));
    DisplayText(Transcript18,FormatValue("\n\t%s","Cost for renovation less than £ 20,000.00"));
    };
  If   renov >= maint And renov < demol And
        (cost + unf) < 20000
   Then
    {
    EconomicAnalysis:Appraisal = Good ;
    DisplayText(Transcript18,FormatValue("\n%s\n","The Decision on economic analysis was based on:"));
    DisplayText(Transcript18,FormatValue("\n\t%s","NPV   for   renovation   less   than   NPV   for   demolition"));
.........................................................................................
.........................................................................................
```

Figure 7.13 gives a partial view of the results of the economic analysis for a given application case. Where a current application case is for a disabled facilities grant or minor works assistance, the system does not perform the economic analysis, instead it proceeds directly to compute the test of resources of the relevant persons to the application.

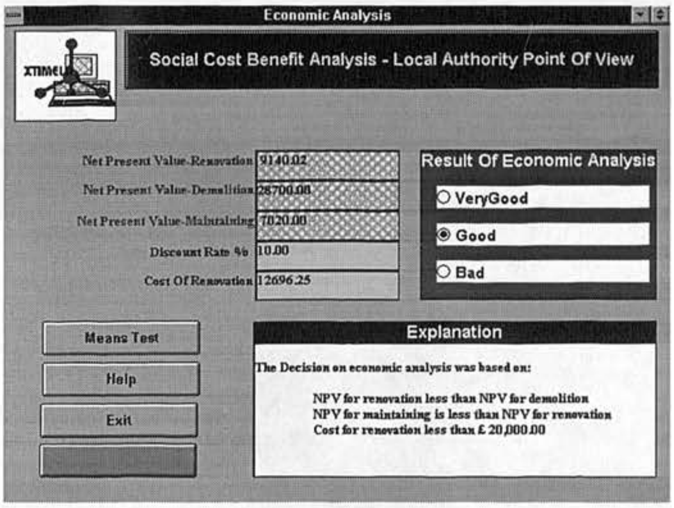


Figure 7.13: A partial view of the results of the economic analysis for a renovation action

7.2.2.8.5- Test of Resources (Means Test)

The system computes the means test using three different procedures corresponding to each category of application cases. It is a complex task which the system computes in reasonable time. The system begins by finding the relevant persons with respect to the current application case. A partial view of how the system implements this step is shown below.

Xtimela-CBR finds the relevant persons

```
..........................................................................................
AskValue(RelevantPersons:Partner);
AskValue(RelevantPersons:IS);
AskValue(RelevantPersons:Education);
AskValue(RelevantPersons:Responsible);
If RelevantPersons:Partner #= Yes
 Then AskValue(RelevantPersons:PartnerAge);
If RelevantPersons:ApplicantAge >= 19 And RelevantPersons:Partner #= No
 Then
 {
  RelevantPersons:RPersons = Applicant;
  RelevantPersons:NumberOfRP = 1;
 };
If RelevantPersons:ApplicantAge <= 19 And RelevantPersons:ApplicantAge >= 16
  And RelevantPersons:IS #= No And RelevantPersons:Partner #= No
 Then
 {
  RelevantPersons:RPersons = Applicant;
  RelevantPersons:NumberOfRP = 1;
 };
If RelevantPersons:ApplicantAge <= 19 And RelevantPersons:ApplicantAge >= 16
 And RelevantPersons:Partner #= No And RelevantPersons:Education #= No
 Then
 {RelevantPersons:RPersons = Applicant;
  RelevantPersons:NumberOfRP = 1; };
..........................................................................................
```

In the next step, Xtimela-CBR computes the applicable amount and the eligible income and capital with respect to each relevant person and with respect to the current application case.

Finally, Xtimela-CBR computes the reduction of grant with respect to the current application case. A partial view of how Xtimela-CBR computes the reduction of grant is presented below.

Xtimela-CBR computes the reduction of grant

```
.......................................................................
SendMessage(ReductionOfGrant,CalTotalDI);
 ResetValue(ReductionOfGrant:Reduction);
 ResetValue(ReductionOfGrant:ReductionOfGrant);
 ResetValue(ReductionOfGrant:Years10);
 ResetValue(ReductionOfGrant:Years5);
 ResetValue(ReductionOfGrant:Previous);
 ResetValue(R:Certificate);
 ResetValue(R:GrantHistory);
  AskValue(R:Certificate);
 AskValue(R:GrantHistory);
 If ReductionOfGrant:TotalDI > 0
  Then
   {
   Let [ disinc ReductionOfGrant:TotalDI]
    {
     If R:Certificate #= Owner_Occupation
      Then
       {
        If disinc <= 47.95
         Then ReductionOfGrant:Reduction = disinc * 17.1;
        If disinc > 47.95 And disinc <= 95.89
         Then ReductionOfGrant:Reduction = 47.95*17.1 + (disinc-47.95) * 34.2;
        If disinc > 95.89 And disinc <= 191.78
         Then ReductionOfGrant:Reduction =47.95*17.1 + 47.94*34.2 + (disinc -95.89) * 136.8;
        If disinc > 191.78
         Then ReductionOfGrant:Reduction = 47.95*17.1 + 47.94*34.2 + 95.89*136.8 + (disinc - 191.78)* 341;
       };
      If R:Certificate #= Tenants
       Then
        {
         If disinc <= 47.95
          Then ReductionOfGrant:Reduction = disinc * 10.28;
         If disinc > 47.95 And disinc <= 95.89
          Then ReductionOfGrant:Reduction = 47.95*10.28 + (disinc-47.95) * 20.56;
         If disinc > 95.89 And disinc <= 191.78
          Then ReductionOfGrant:Reduction = 47.95*10.28 + 47.94*20.56 + (disinc -95.89)*82.22;
         If disinc > 191.78
          Then ReductionOfGrant:Reduction = 47.95*10.28 + 47.94*20.56 + 95.89*82.22 + (disinc - 191.78)* 205.55;
        };
      };
   }
   Else ReductionOfGrant:Reduction = 0;
  If R:GrantHistory #= Yes
   Then
    {
     AskValue(ReductionOfGrant:Years10);
     AskValue(ReductionOfGrant:Years5);
     If ReductionOfGrant:Years10 #= Yes Or
        ReductionOfGrant:Years5 #= Yes
      Then
       {
```

```
    AskValue(ReductionOfGrant:Previous);
    Let[ x ReductionOfGrant:Previous]
      [ red ReductionOfGrant:Reduction]
     ReductionOfGrant:ReductionOfGrant = red - x ;
    };
  }
  Else ReductionOfGrant:ReductionOfGrant = ReductionOfGrant:Reduction ;
DisplayText(Transcript42,FormatValue("%9.2f",ReductionOfGrant:Previous));
DisplayText(Transcript40,FormatValue("%9.2f",ReductionOfGrant:ReductionOfGrant));
};
..........................................................................
..........................................................................
```

Figure 7.14 gives a view of the results of the test of resources carried out according to the legal framework for an application case with only one relevant person who is an employed earner of 45 years of age without children.

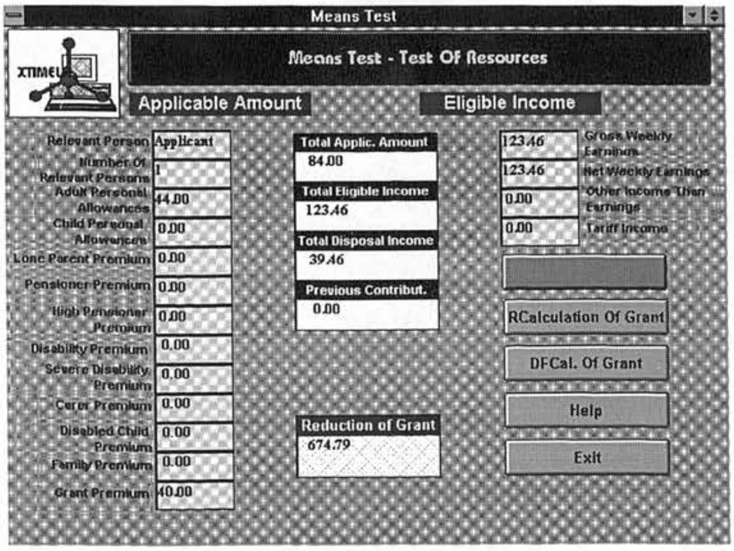


Figure 7.14: A view of the results of the mean test

7.2.2.8.6- Amount of Grant

After completing the test of resources, Xtimela-CBR computes the amount of grant to be awarded and the non-granted amount with respect to the current application case using an algorithm. Given the cost of works and the reduction in the amount of grant, the system computes the amount of grant to be awarded.

7.2.2.8.7- Degree of Eligibility of the Current Application

The eligibility of an application is a compound index, which is result of the computation of other attributes. The degree of eligibility is a fairly reliable index for guiding the retrieval of a solution for the current application case. Xtimela-CBR incorporates three different procedures corresponding to each category of applications cases. A partial view of how the system computes the degree of eligibility is shown below.

Xtimela-CBR computes the degree of eligibility using a criteria

```
..........................................................................
If Global:GrantType #= "Renovation Grant"
 Then
 {
  SendMessage(R,CalEligibility);
  ShowImage(StateBox1);
  SetImageLink(StateBox1,R,Eligibility);
  SetValue(Global:Eligibility,R:Eligibility);
 };
 If Global:GrantType #= "Disabled Facilities Grant"
 Then
..........................................................................
..........................................................................
AskValue(Self:Completion);
 AskValue(Self:HousingNeeds);
 AskValue(Self:HouseCondition);
 AskValue(Self:Cert);
 AskValue(Self:Needs);
 AskValue(Self:GrantHistory);
 If Self:Fitness #= UnfitForHumanHabitation And
    Self:PreliminarAssessment #= "Proceed With Application"
  Then
  {
   If Self:Completion #= Yes And
      Self:Appraisal #= VeryGood  Or
      Self:Appraisal #= Good And
      Self:HouseCondition #= Worst_condition And
      Self:HousingNeeds #= High And
      Self:Cert #= Yes And
      Self:GrantHistory #= No And
      Self:GrantTotal > 0 And
      Self:Needs #= Greatest_housing_needs Or
      Self:Needs #= Unable_to_find_solution Or
      Self:Needs #= Require_special_care_and_support Or
      Self:Needs #= Have_dependent_children Or
      Self:Needs #= Vulnerable_people Or
      Self:Needs #= Have_mental_illness_or_handicap Or
      Self:Needs #= Unintentional_homless Or
      Self:Needs #= Discharged_from_a_hospital_to_be_closed Or
      Self:Needs #= Works_required_urgently
     Then Self:Eligibility = VeryHigh;
   If Self:Completion #= Yes And
      Self:Appraisal #= VeryGood  Or
      Self:Appraisal #= Good And
      Self:HouseCondition #= Rest_condition And
      Self:HousingNeeds #= Normal And
      Self:Cert #= Yes And
      R:GrantHistory #= No And
      Self:GrantTotal > 0  And
      Self:Needs #= Normal And
      Self:Appraisal #= VeryGood Or R:Appraisal #= Good
     Then Self:Eligibility = High;
   If Self:Completion #= Yes And
```

```
    Self:Appraisal #= Good And
    Self:HouseCondition #= Rest_condition And
    Self:HousingNeeds #= Satisfied And
    Self:Cert #= Yes And
    Self:GrantTotal > 0  And
    Self:Needs #= Normal And
    R:GrantHistory #= No Or R:GrantHistory #= Yes
   Then Self:Eligibility = Medium;
 If Self:Completion #= Yes And
    Self:Appraisal #= Bad And
...........................................................................
...........................................................................
```

Xtimela-CBR evaluates the eligibility considering the criteria described in chapter 5. This criteria was designed taking into account the legal framework and the views of several human experts in the HRGS domain. Figure 7.15 gives a view of the computed eligibility found for a current application case.

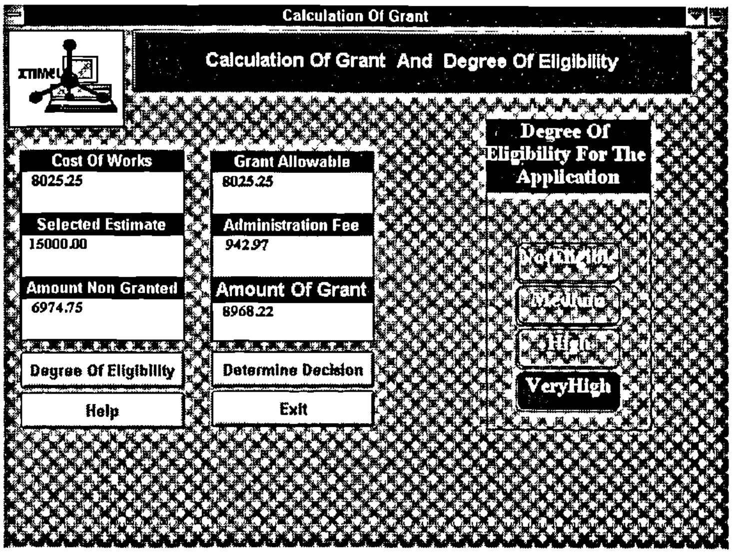


Figure 7.15: A view of the computed eligibility found for a current application case

7.2.2.8.8- The Decision for the Current Application Case

To find a decision (solution) for the current application the system's control invokes the case-base retriever. The case-base retriever derives a decision for the current application by searching and matching a similar case in the case library following the process described in section 7.2.2.6.1. A partial view of how the case-base retriever derives the decision is shown below. The case-base retriever performs the current task by: i) computing the indices for the new application case; and ii) deriving a solution from application cases stored in the case library.

Xtimela-CBR begins the retrieval process by indexing the new application case

```
...........................................................................
{
 If Null?(Global:Eligibility)
```

```
 Then PostMessage("First determine the degree of eligibility of the application. Press button Eligibility");
HideWindow(Session10);
ResetWindowMenu(Session10);
ShowWindow(Session11);
MaximizeWindow(Session11);
RemoveWindowMenu(Session11);
ClearTranscriptImage(Transcript49);
ClearTranscriptImage(Transcript50);
ClearTranscriptImage(Transcript51);
ClearTranscriptImage(Transcript52);
DisplayText(Transcript49,FormatValue("\t%s",Global:GrantType));
If Global:GrantType #= "Renovation Grant"
 Then
  {
   SendMessage(R,CalGrantDecision);
   SendMessage(R,Report);
  };
If Global:GrantType #= "Disabled Facilities Grant"
 Then
..............................................................................
..............................................................................
{
PostBusy(ON,"Determining The Grant Decision Plan");
Wait(1);
PostBusy(OFF);
SendMessage(Self,CalCondition);
SendMessage(Self,CalIncome);
ResetValue (Self:FutureOfArea);
ResetValue(Self:AdjProperties);
AskValue (Self:FutureOfArea);
Let [ area Self:FutureOfArea ]
 {
  If area #= To_be_declared_renewal_area And
     Self:Eligibility #= VeryHigh Or
     Self:Eligibility #= High Or
     Self:Eligibility #= Medium
    Then SendMessage(Self:DecisionPlan);
 If area #= None
   Then
    {
     AskValue(Self:AdjProperties);
     If Self:AdjProperties #= None Or
        Self:AdjProperties #= Concentration_of_poor_housing Or
        Self:AdjProperties #= Area_had_not_significant_grant_aid_work And
        Self:Eligibility #= VeryHigh Or
        Self:Eligibility #= High Or
        Self:Eligibility #= Medium
      Then SendMessage(Self,DecisionPlan);
     If Self:AdjProperties #= Area_had_significant_grant_aid_work And
        Self:Eligibility #= Medium  Or
        Self:Eligibility #= Low
..............................................................................
..............................................................................
```

Then, the case-base retriever derives the decision (solution)

```
..............................................................................
PostBusy(ON,"Consulting The Case DataBase Of Past Applications");
Wait(1);
```

```
PostBusy(OFF);
ClearList(Global:CaseBase);
ResetValue(Self:ConstructionEra);
ResetValue(Global:Case);
AskValue(Self:ConstructionEra);
If Self:ConstructionEra #= Pre_1919
 Then
  {
  GetInstanceList (Pre_1919 , Global:CaseBase);
  Global:Case = SelectList(Global:CaseBase,case,
      {
      case:Eligibility #= Self:Eligibility And
      case:Income #= Self:Income And
      case:Condition #= Self:Condition And
      case:Fitness #= Self:Fitness
      });
  If Not(Null?( Global:Case))
   Then
   {
    Let [ cas Global:Case ]
    {
     SetValue(Self:GrantDecision,cas:GrantDecision);
     SetValue(Self:GrantCondition,cas:GrantCondition);
     SetValue(Self:WorksCondition,cas:WorksCondition);
     SetValue(Self:Note,cas:Note);
    };
   }
   Else NewSolution();
  };
If Self:ConstructionEra #= Pos_1919
 Then
........................................................................
........................................................................
```

7.2.2.9- Domain Models

7.2.2.9.1- Building Model

The building model is represented in Xtimela-CBR as a *part-whole* object hierarchy, where instances represent building components. Objects are composed of: i) slots representing building components features; and ii) methods representing building component functions and behaviours.

7.2.2.9.2- Standard of Fitness Model

The standard of fitness model is represented in the Xtimela-CBR as an object hierarchy, where: i) classes represent the requirements of the fitness standard; and ii) instances represent building components associated with those requirements. Figure 7.16 shows a partial view of the standard of fitness model as represented in Xtimela-CBR knowledge-base.

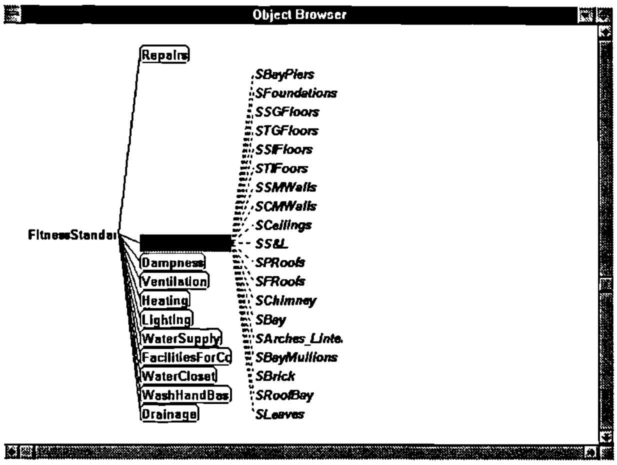


Figure 7.16: A partial view of the standard of fitness model

7.2.2.9.3- Grant Conditions Model

The grant conditions model is represented in the Xtimela-CBR as an object hierarchy, where: i) classes represent the types of grant; and ii) instances represent conditions of eligibility associated with each type of grant. Figure 7.17 shows a partial view of the grant conditions model as represented in Xtimela-CBR knowledge-base.

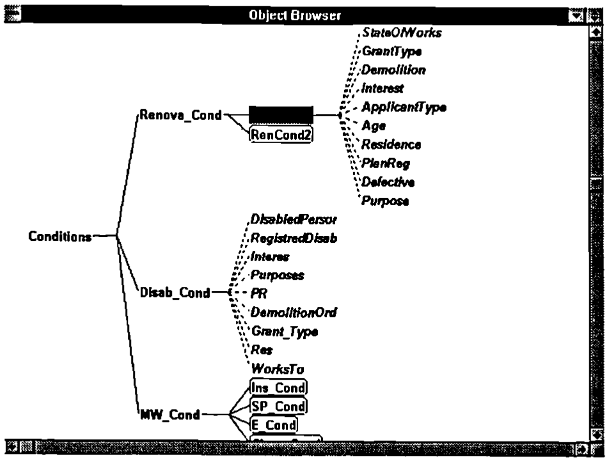


Figure 7.17: A partial view of the grant conditions model

7.2.2.9.4- Disabled Facilities Model

The disabled facilities model is represented in the Xtimela-CBR as an object hierarchy, where: i) classes represent the requirements for carrying out the adaptations; and ii) instances represent specific conditions associated with each requirement.

7.3- LESSONS LEARNED FROM THE IMPLEMENTATION

In implementing Xtimela-CBR, a number of practical lessons were learnt about building the system using the architecture outlined in chapter 6. They were grouped into the following main headlines.

Lesson 1: Not every real-world application case is complete

Although the domain of the assessment of applications for the HRGS contains a large number of application cases, not every one of them is well-documented. The condition survey reports are often incomplete or inconsistent. In such cases, the level of unforeseen works found after the conclusion of works reflects such problems. Due to the Client's policy, only mandatory application cases are available for renovation and disabled facilities grants. No discretionary grant applications for those types of grants have been approved in recent times. This situation raised several constraints which had to be taken into account during the implementation, such as:

- the need to ensure the validity of every application case added to the case library;
- how to use an incomplete past application case to support decision-making in the domain; and
- how to cover other areas of domain for which past application cases were not available.

To address the first two issues, the following was adopted: i) every application case added to the case library was validated, by running it as new problem; and ii) partially documented application cases were included in the case library in the same proportion as the full documented ones. To address the third question, one possible solution is to include prototype discretionary application cases in the case library. Such prototypes can be used to build solutions to new problems where real-word application cases do not exist.

Lesson 2: Real-world application cases are large in terms of the information held

A well-documented past grant application contains a large amount of potentially useful information about applicants, properties, application goals and constraints, post-works, feedback analysis, and so on. This presents special requirements for representing, indexing and presenting an application case. These questions can be addressed by using: i) a robust structure for representing an application case; ii) a flexible indexing format; iii) a more structured application case library; and iv) a format to present the application that enables the user to navigate both within and across the application case.

Lesson 3: The number of potentially useful application cases is large

The Client stores a large number of past grant applications which are potentially useful for solving new problems. They store a large body of context related knowledge which needs to be organised and made easily accessible for the Client's experts. Besides that, the number of applications for grants is growing and it was over 50 per month last year. Thus, the case library must be large enough to accommodate the number of potentially useful application cases. This presents problems of how to organise a huge number of application cases in the case library.

Lesson 4: The availability of past cases which provide specific knowledge is vital for the success of a project using CBR

The development of Xtimela-CBR has shown that one of the things that must be considered in building a system using CBR is whether or not past cases containing specific knowledge are available in the domain. If these cases are not easily available or are very expensive to collect building a system will be difficult. In problem solving tasks, cases must provide knowledge related to specific experiences which can be used to build solutions for new problems in similar situations.

Lesson 5: CBR offers several advantages in knowledge acquisition and representation

Knowledge acquisition was simple and quick for Xtimela-CBR. Once the case report formats were found it was easy to translate the information contained in the past grant application into full-text case reports. After the initial seed application cases were validated, they were immediately useful for allowing the system to work even with an incomplete knowledge-base. With the seed cases, the system was able to acquire additional past application cases by processing and validating them as new problems. Representing application cases as objects and using the object inheritance simplified the implementation of the application case database.

Lesson 6: The system must present relevant information through the consultation and enable the user to modify it anytime

Due to the time spent in attending to the aims of this research, little time was allocated to the user interface. Therefore, the version of Kappa-PC used does not yet provide enough tools to quickly build highly interactive and dynamic user interface. The assessment of a grant application is an information-processing intensive task which requires a highly interactive user interface. Besides, the system must provide the user with the relevant information in the right format and at the right time. Integrating the application with other presentation software could be a viable solution.

7.4- SUMMARY AND CONCLUSIONS

This chapter presented a general overview of the implementation of Xtimela-CBR as an application using Kappa-PC tool. Xtimela-CBR has reasonably successfully demonstrated how an architecture integrating several methods and a case library can be used for assessing applications for the HRGS. The implementation of Xtimela-CBR using Kappa-PC in context of the research aims and scope can be considered reasonably successful. Xtimela-CBR has accomplished the main goals of this research and it has enabled the author to investigate the validity of CBR models in the assessment of applications for the HRGS. Several problem solving approaches unified into a single framework were used for implementing Xtimela-CBR.

The vocabulary chosen to represent and index application cases is stable and expressive enough with respect to the tasks where CBR is applied. However, the current version of Xtimela-CBR is only a partial demonstration of how a hybrid architecture can successfully support the assessment of grant applications. Some of the contributions of Xtimela-CBR are the lessons learned from its limitations. The final version of Xtimela-CBR is a fairly large application in terms of knowledge represented and variety of structures used to implement that knowledge. The performance of the system in terms of time savings and accuracy is quite good: the system usually takes among 30 and 60 minutes to assess a grant application where the input data is readily available.

In the next chapter, the process of verifying and validating the system will be described and discussed. Some of the conclusions attained during validation are related to the way in which Xtimela-CBR was implemented.

CHAPTER 8

VERIFICATION AND VALIDATION OF THE SYSTEM

8.1- INTRODUCTION

The description of Xtimela-CBR system was outlined and discussed in the last chapter. In this chapter special attention is devoted to a description of the work carried out for the verification and validation (V&V) of Xtimela-CBR.

According to Addis, (Addis T. R., 1993) evaluating a KBS involves assessing: i) the utility or viability of a hypothesis, given a particular purpose; and ii) the overall effectiveness of the KBS in addressing the task(s) for which it was built. Kolodner (Kolodner J., 1993) pointed out that the quality of a system using CBR can be measured by: i) the experiences it has in its case library; ii) its ability to understand new situations in terms of old experiences; iii) its adeptness at adaptation; and iv) its ability to integrate new experiences into its library appropriately.

Taking into account the definition provided by Addis (Addis T. R., 1993), the Kolodner's (Kolodner J., 1993) statement about the quality requirements for a CBR system and the aims of this research, the V&V of Xtimela-CBR consisted of three main objectives:

1. To assess the utility and the viability of the research hypotheses.
2. To assess the overall effectiveness of the system, as a whole and by its sub-components, in addressing the assessment of applications for the HRGS as specified by the task structure presented in chapter 4.
3. To assess the qualities and benefits of the system in addressing the Client's needs.

On the basis of these objectives, a concise V&V plan for Xtimela-CBR was designed and implemented. The Client provided much of the support needed for this plan.

8.2- V&V OF KBSs

8.2.1- DIFFICULTIES IN THE V&V KBSs

There are a number of difficulties in verifying and validating KBSs during the development life cycle. This was recognised by several researchers working in the field. The difficulties in verifying and validating KBSs according to several authors include (O'Keefe R. et al., 1987; Green C. and Keyes M., 1987; Nasser J., 1988; Gupta U., 1991; and Cervera E., 1993):

- The lack of an uniform terminology. A variety of terms are used to define the evaluation of KBSs.
- The lack of detailed and testable requirements and specifications.
- The area is still far from having a standard set of tools and techniques for V&V covering the complete KBS life cycle.
- What to validate? Any intermediate results, the final result, the reasoning of the system, or any combination of these three.
- What to validate against? Validate against known results as well as against expert performance.
- What to validate with? In the real world, some times, only a small sample of test cases is available. Further, the choice of test cases biases the success of any validation.
- How to control the cost of the V&V? V&V can be time consuming and expensive?
- How to control bias? When judging KBS performance, an expert biased against introducing computer-based systems may assess the system unfairly?

Additionally, regarding the V&V of CBR systems, it was found that:

- Most of the existing methods are designed for the V&V of rule-based systems where the knowledge-base is functionality static. A CBR system after each problem solving session can add a new case to its case library, and therefore inherently change the functionality of the overall system. To deal with these functional changes requires different methods from those available for the rule-based systems (Hennessy D. and Hinkle D., 1992).
- How to validate a case library storing past cases which have been used before and they are assumed to be useful in future for similar conditions.

8.2.2- CITED APPROACHES TO THE VERIFICATION OF KBSs

A literature review about the subject has shown that the available tools and techniques for verifying KBSs were designed for rule-based and frame-based systems. Thus, most of these tools and techniques are logic-based.

The main features of the major approaches to the verification of KBSs include:

- Nasser J. (Nasser J., 1988) addresses the qualities which should be tested during the verification process. These qualities include: the adequacy of the knowledge representation; the validity, consistency, and completeness of the knowledge-base; the explanations provided by the system for its decisions; and the accuracy and consistency of the reasoning mechanisms in solving domain problems.
- Preece (Preece A., 1990) specified the verification of KBSs as checking for: conflicting, redundant and subsumed sets of rules; conflicting, redundant and subsumed inference chains; cyclic inference chains; useless rules; dead-end rules; unsatisfied conditions; missing values; missing rules; and syntax errors.
- According to Gupta (Gupta U., 1991) verification is designed to determine if the system completely and accurately implements user specifications (i.e. it determines if the system was built right).
- Hoppe and Meseguer (Hoppe T. and Meseguer P., 1993) specified the verification of KBSs as verifying: the knowledge base, the inference engine, the user interface, the input/output behaviour and any other functional aspect.
- Jafar and Bahill (Jafar M. and Bahill A., 1993) developed the VALIDATOR which is a tool designed to verify rule-based and frame-based systems. This program performs the verification by checking for: syntactic and semantic errors; unused rules, facts and questions; redundant constructs; rules that use illegal values; incorrectly used instances; and multiple methods for obtaining values for expressions.

8.2.3- CITED APPROACHES TO THE VALIDATION OF KBSs

Taking into account the validation requirements of the Xtimela-CBR, some of the available approaches were reviewed, including those based on test cases. A general description of these approaches is presented as follows:

Face validation: Face validation is a preliminary approach to validation which consists of assessing the face value of a KBS, with regard to a prescribed acceptable performance range. Thus, project team members, potential expert system users, and domain experts subjectively compare system performance with human expert performance (O'Keefe R. et al., 1987).

Predictive validation: Predictive validation consists of evaluating the system performance using historic test cases. Thus, the validation of a KBS is driven by past input data from the test cases, and its results are compared with those obtained from the cases or provided by human experts (O'Keefe R. et al., 1987).

Field tests: Field tests place the system in the field, and then seek out perceived performance errors as they occur, by using real cases. From the developer's viewpoint, this offers two considerable advantages: Firstly, it places the burden of testing upon users. Secondly, acceptable performance ranges are obtained implicitly, since users may cease to report problems when acceptable performance ranges are reached (O'Keefe R. et al., 1987).

Subsystem validation: Subsystem validation requires that the KBS be decomposed into subsystems, enabling the performance of each subsystem to be observed under given input data. In this approach, subsystems are validated one at a time as they are developed (O'Keefe R. et al., 1987).

Visual interaction: Visual interaction validation is based on the visual animation of the KBS working. It has been successfully employed in validating operational research models (O'Keefe R. et al., 1987).

Sensitive analysis: Sensitive validation is performed by systematically changing KBS input variable values and parameters over some range of interest and observing the effect upon system performance. It is especially useful where few or no historical test cases are available (O'Keefe R. et al., 1987).

Robustness: Robustness test is performed by using specially selected cases which reflect extreme conditions under which the system may be operated. The test cases should be selected carefully (Marcot B., 1987).

Knowledge-base refinement: Knowledge-base refinement considers the improvement of a KBS from a set of cases with known solutions. This set of cases should be a representative sample of the problem domain. The goal of the knowledge-base refinement is to improve KBS validity, that is to say, the KBS should be more valid after implementing each single refinement in the knowledge-base. To increase KBS validity, knowledge-base refinement should be guided by error importance with respect to the KBS task. Most serious errors should be solved at first, possibly causing some errors of lower importance, but always assuring a net validity gain (Meseguer P., 1993).

Marcot (Marcot B., 1987), Klein and Brezovic (Klein G. and Brezovic C., 1988) proposed a set of factors for evaluating the performance of a KBS. These factors include:

- Accuracy: The accuracy is measured by comparing the results provided by the system with historic (known) data and observing the correctness of the outcome.
- Adaptability: The adaptability is measured by the possibilities for future development of the system.
- Depth: The depth is measured by the range of conditions the system will address.
- Generality: The generality is measured by the capability of a system to be used in a broad range of similar problems.
- Usefulness: The usefulness validates that the system contains necessary and adequate parameters and relationships for use in various contexts.
- System effectiveness: The system effectiveness means the ability of the system to perform the task it is assigned, and the efficiency with which it carries out these tasks. This involves testing the following qualities: i) content knowledge; ii) power (time to solution, success rate and quality); iii) human operator equivalent; iv) flexibility; and v) expandability.
- User effectiveness: The user effectiveness addresses the way in which the system enhances the user's capabilities.
- Organisational effectiveness: The organisational effectiveness measures how the introduction of the system will affect the performance of the organisation.

8.3- A FRAMEWORK TOWARDS THE V&V OF Xtimela-CBR

8.3.1- DIFFICULTIES IN THE V&V OF Xtimela-CBR

In addition to some of the limitations relating to the V&V of KBSs mentioned above, there are several practical constraints which are unique to the system developed during this research. They include the following limitations:

1. Since the general aim of this research is to investigate and explore the viability and suitability of using CBR in the domain of the assessment of applications for the

HRGS, a great part of the time available for the research was committed to the knowledge modelling, case acquisition and programming tasks. As a result, the time available for the V&V was limited.

2. Evaluating Xtimela-CBR required a number of representative real application cases stored in its case library and a number of real test cases to validate against. This required the commitment of the Client. To get the Client's involvement in the project required spending considerable time in obtaining contacts and holding meetings.
3. The past grant applications are stored in files. These files contain confidential documents with very restricted circulation. This specific limitation resulted in additional time spent in handling such files.
4. Because of the confidentiality and property pertaining to the knowledge stored in the system's case library, it was not possible to invite more experts from outside the Client to participate in the validation process.
5. The Xtimela-CBR system is the only one of its kind in the HRGS domain. Although the author does not rule out the possibility that some similar system may have been developed or may be still under development. Little research work has been published on this domain. This lack of similar research or systems, eliminated the possibility of validating Xtimela-CBR against existing comparable systems.
6. The HRGS came into force in the middle of 1990 and since then it has been changed several times. The present research started at the end of 1991 and had to incorporate those successive modifications of the HRGS. As a consequence, the proposed system also had to be altered in due course. This delayed the V&V of the Xtimela-CBR and reduced the time available for it.

8.3.2- ADVANTAGES OF V&V OF Xtimela-CBR

Although there are a number of limitations in the V&V of the system, there are also a number of advantages which helped the evaluation process. These supporting issues pertain to three types of reasons: i) the specific nature of the domain; ii) the modelling work carried out in the context of this research; and iii) the shell used for implementing the system. They are presented as follows:

Due to the specific nature of the domain:

- The number of test cases available for validation purposes was large.
- The test cases are organised in different standard formats according to the type of grant sought. This organisation facilitated the validation of the system both as a whole and by its components.

Due to the modelling work:

- Carrying out the task analysis helped to overcome the limitation of the lack of testable requirements and specifications. The task structure for the assessment of grant applications provided a good specification for the system. This specification allowed to evaluate the system not just in terms of its tasks and goals, but also in terms of its inferences. Yen and Lee (Yen J. and Lee J., 1993) stressed that by organising a specification of a KBS around the task structure supports and makes easy its V&V.

Due to the development shell:

- Kappa-PC provides some tools for verifying knowledge-bases. These tools include: the syntax debugger; and the semantics checker.

8.3.3- THE FRAMEWORK FOR THE V&V

The V&V of Xtimela-CBR was incorporated into the system implementation cycle. The system was verified and validated at every implementation stage of the CCA method. Marcot (Marcot B., 1987) stressed that the V&V of KBS should be integrated into the system development life cycle. Each validation step of the system with a set of test cases was followed by the refinement and updating of the system guided by detected errors and malfunctions found at that step. Therefore, each case added to the case library was verified and validated. The goal was that after a validation step an improvement of the system should be obtained through a refinement. Meseguer (Meseguer P., 1993) pointed out that KBS validation supported by knowledge-base refinement should increase the system's validity with respect to a task for which it was built. Although the literature review in the area of V&V has shown that there were a few software tools for V&V of KBSs, e.g., Validator (Jaffar M. and Bahill T., 1993), Improver (Meseguer P., 1993), and Expert-System Checker (Cragun J. and Steudel H., 1987), it was decided to use only the tools provided by Kappa-PC combined with manual testing using real test application cases. According to Meseguer (Meseguer P., 1993) an important part of the validation process of KBSs relies on manual testing using known cases.

Taking into account the limitations and advantages described in sections 8.3.1 and 8.3.2 and to the objectives attached to V&V of the system, a framework to evaluate Xtimela-CBR was developed and implemented. This framework includes some of the methods introduced in sections 8.2.3 and 8.2.4 which are summarised in tables 8.1 and 8.2.

Table 8.1: Summary of methods used for the verification of Xtimela-CBR

Method	Techniques
Correctness checking	Check for spelling, syntactic and semantic errors.
Consistency checking	Check inconsistent *if then* expressions; Check redundant *if then* expressions; Check conflicting *if then* expressions; Check for subsumed *if then* expressions; Check unnecessary "if" conditions in *if then* expr.; Check inconsistent retrieval functions; and Check inconsistent case features.
Completeness checking	Check completeness of case library; Check completeness of application cases; Check missing objects and object attributes; Check missing links; Check un-referenced attribute values; Check illegal attribute values; and Check missing *if then* expressions;

A detailed description of the V&V framework including its implementation is outlined in later sections. Methods which were introduced in sections 8.2.3 and 8.2.4 but not used for the V&V of Xtimela-CBR include: i) visual animation; and ii) sensitive analysis. Visual animation was excluded because of the limitations of the graphic facilities provided by Kappa-PC, which prohibit the visual animation of the system. Sensitivity analysis was excluded, because the number of test cases available for predictive validation and field tests were large and representative of the current demands.

Table 8.2: Summary of methods used for the validation of Xtimela-CBR

Method	Techniques
Validation as process of system implementation	Validation of the case library; and Validation of case-base retrieval.
Predictive validation	Performance and functional features; Validation against criteria; Sub-system validation; and Intermediate and final results.
Field tests	System effectiveness; and User effectiveness.
Face validation	User/Client acceptance; Test of innovative features; and Test ergonomic factors.
Robustness test	With hypothetical application cases reflecting extreme conditions.
Knowledge-base refinement	Integrate validation with refinement.

8.4- DESCRIPTION OF THE V&V OF Xtimela-CBR

The V&V of Xtimela-CBR required a set of valid application cases stored in its case library. The V&V process started with the validation of the 60 seed application cases stored in its case library and it finished when the case library had over 130 application cases. Thus, it was possible to carry out a compressive evaluation of Xtimela-CBR within the objectives of this research and to conclude about its benefits and weaknesses.

8.4.1- VERIFICATION OF Xtimela-CBR

As listed in table 8.1, the verification of Xtimela-CBR consisted of checking: i) the correctness of the programme; ii) the consistency of the programme; and iii) the completeness of the system components at each stage and between each stage of the implementation cycle.

According to Preece (Preece A., 1990), the consistency and completeness checking are both methods usually used to determine the self-consistency and completeness of rule-based systems where the rule is assumed to be a logic expression.

Xtimela-CBR system has no rules represented as independent knowledge structures. Instead it uses functions and methods, some of them incorporating *if then* expressions. Thus the consistency and completeness methods were applied to verify the logic expressions defined in functions and methods.

8.4.1.1- Checking Correctness Of The Programme

Checking the correctness consisted essentially of:

1. Checking the syntax of the knowledge represented in the system by using the debugging tools provided by Kappa-PC.
2. Checking the spelling by using the spelling tool provided by the Winword editor throughout all of the programme.
3. Checking the underlying semantics of the object hierarchies.

Following the above approach, every function and method in the system was scanned and checked for possible syntax errors. The Kappa-PC debugging tools call attention to possible errors allowing to amend the syntax and semantic errors. After the program had been checked for syntax and semantic errors, it was converted into a text file later and edited for spelling in the Winword editor. The Winword editor allows to the detection and correction of spelling errors contained in the programme. The underlying semantics of object hierarchies have been continually checked by inspecting the correctness of object links and object structures in the Xtimela-CBR knowledge-base.

8.4.1.2- Consistency Checking

According to Preece (Preece A., 1990), checking consistency involves checking for: inconsistent, redundant and subsumed rules; and cyclic inferences. Nguyen (Nguyen T. et al., 1987) pointed out that consistency checking is usually performed by simple comparison of individual rules. Verifying the consistency in the Xtimela-CBR knowledge-base consisted of checking: i) the consistency of the case library; ii) the consistency of the retrieval and matching procedures; iii) the consistency of adaptation procedures; and iv) checking consistency of other procedures. The main building blocks of these components in Kappa-PC language are: objects; functions; and methods defined in the objects. Thus, consistency has been checked by a continual process of detecting, removing and correcting:

- The functions and methods where the following consistency problems may exist: inconsistent *if then* expressions; redundant *if then* expressions; conflicting *if then* expressions; subsumed *if then* expressions; unnecessary if conditions *if then* expressions.
- The case memory, where the following consistency problems may exist: inconsistent retrieval functions; inconsistent case features and indexes.

<u>Inconsistent *if then* expressions</u>

For rule-based systems, according to Preece (Preece A., 1990), a rule set is consistent only if there is no way the rules can assert a contradiction from valid input. In Xtimela-CBR, every *if then* expression has been checked individually, and as a set for any such contradictory results. An example of an inconsistent *if then* expression is given by following expressions:

$$a \wedge b \rightarrow c \quad (1)$$

$$a \wedge b \rightarrow \neg c \quad (2)$$

Expression 1 is inconsistent with expression 2. Checking the inconsistency of an *if then* expression consisted in inspecting the *if* condition and the *then* conclusion.

Redundant *if then* expressions

According to Nguyen (Nguyen T. et al., 1987), a rule is redundant if it succeeds for the same input and has the same conclusion as another rule. Although redundancy does not necessarily cause logical problems, it might affect the system's performance (Suwa M. et al., 1982). In Xtimela-CBR, every set of *if then* expressions have been checked for redundant expressions. An example of a redundancy is given by following expressions:

$$a \rightarrow b \quad (4)$$

$$b \rightarrow c \quad (5)$$

$$a \rightarrow c \quad (6)$$

Expressions 4 and 5 are redundant with expression 6.

Conflicting *if then* expressions

Two rules are conflicting if both succeed for the same input but with different conclusions. Conflicting rules cause logical problems and affect the system results. In Xtimela-CBR, every set of *if then* expressions have been checked for conflicting expressions. An example of conflicting *if then* expressions is given by following expressions:

$$a \rightarrow b \quad (7)$$

$$a \rightarrow c \quad (8)$$

Expressions 7 and 8 are conflicting in the expression set.

Subsumed *if then* expressions

According to Preece (Preece A., 1990), one rule is subsumed by another if the two rules have the same conclusions, but one contains additional conditions for the situation in which it will succeed. In Xtimela-CBR, every set of *if then* expressions has been checked for subsumed expressions. An example of subsumed *if then* expressions is given by following expressions:

$$a \wedge b \rightarrow c \quad (9)$$

$$a \rightarrow c \quad (10)$$

Rule 9 is subsumed by rule 10.

Unnecessary *if* conditions

According to Preece (Preece A., 1990), two rules contain unnecessary *if* conditions when they: i) succeed with the same conclusion; ii) the *if* condition in one rule is conflicting with an *if* condition in other rule; and iii) all other *if* conditions in both rules are equivalent. An example of unnecessary *if* conditions are given by the following expressions:

$$a \wedge b \wedge c \rightarrow d \quad (11)$$

$$e \wedge f \wedge g \rightarrow d \quad (12)$$

where $a \equiv e$, $b \equiv f$ and $c \neq g$.

In Xtimela-CBR, every set of *if then* expressions has been checked for unnecessary *if* conditions.

Inconsistent retrieval functions

At an early implementation stage the retrieval functions retrieved unrelated application cases. To detect and avoid this problem, Xtimela-CBR has been checked for: i) inconsistent, and conflicting index features; and ii) inconsistent and conflicting matching functions. These checks were carried out by running a set of test application cases and testing each system's component. All problems found have been removed or corrected.

Inconsistent case features and indexes

Consistency of the case library has been checked for: i) inconsistent, conflicting or redundant application case features; and ii) inconsistent and conflicting values of the application case features. These checks were carried out by running a set of test application cases. All problems found have been removed or corrected.

The above checks were undertaken systematically during the implementation cycle of Xtimela-CBR, using both computer and manual tools.

8.4.1.3- Completeness Checking

According to Preece (Preece A., 1990), a rule base is complete if and only if it can cope with all possible situations that can arise in its domain. Kolodner (Kolodner J., 1993) pointed out that the quality of a CBR system is in part measured by the range of experiences stored in its case memory. Taking in account these two statements, checking the completeness of the Xtimela-CBR system meant having to check the integrity of its main knowledge-base components: i) the application case data base; ii) the domain models; and iii) the functions and methods. All checks described in this section were carried out by running a set of test application cases covering a wide range of problems regarding the occupants and properties.

The completeness checking was carried out mainly:

- For the application case data base: i) checking the completeness of the case library; and ii) checking the completeness of the application cases.
- For the domain models: i) checking for missing objects and missing object attributes; and ii) checking for missing links in the object hierarchy.
- For functions and methods: i) checking for un-referenced attribute values; ii) checking for illegal attribute values; and iii) checking for missing expressions.

Checking the completeness of the case library

Verifying the completeness of the Xtimela-CBR case library consisted of checking its representativeness and integrity. As mentioned before, the V&V of Xtimela-CBR required a set of seed cases in order to guarantee a minimum level of system's functionality. For each application case added to the case library a completeness checking was carried out. The result of each check was then used to guide the selection of the next application case to be added to the case library, with the goal of continually improving the completeness and integrity of the case library.

Checking completeness of application cases

The representation of an application case in the Xtimela-CBR case library is provided by an object which comprises a set of features and methods. Thus, the application cases were checked for:

- missing features and methods;
- lack of representative features; and
- redundant features and methods.

After each checking cycle, the case representation structures were refined by adding new features and/or replacing or deleting old features.

Checking for missing objects and missing object attributes

Domain models in Xtimela-CBR are represented by object hierarchies. The completeness of domain models is assured by the objects represented and the links between objects in the hierarchy.

Objects in domain models capture the knowledge required by different procedures represented as functions and methods. These functions and methods are able to reason by using the knowledge represented in the domain models. Thus, domain models have been checked for missing objects and object attributes by running the set of functions and methods that relate to each domain model. Each checking cycle was followed by the refinement of the domain model checked.

Checking for missing links in the object hierarchy

Brachman (Brachman R., 1985) pointed out that checking the completeness of frame-based systems typically involves determining any missing links in the frame taxonomies. Objects in the domain models are connected by links that represent *part_of* relations and inheritance behaviours. Thus, all domain models were checked for missing links by running the set of functions that relate to each model. Each checking cycle was followed by the refinement of the domain model checked.

Checking for un-referenced feature values

According to Nguyen (Nguyen T. et al., 1987), out of range values occur when the set of possible values of an object's features are not covered by any of the *if* condition of the rules.

Un-referenced feature values were checked by running each set of *if then* expressions that refers or to require information represented in the object features.

Checking for illegal feature values

According to Nguyen (Nguyen T. et al., 1987), an illegal attribute value occurs when a rule refers to an attribute value that is not in the set of legal values. Illegal values in functions were checked and corrected by using the tools provided by the Winword for Windows editor.

Checking for missing *if then* expressions

Checking for missing *if then* expressions was carried out by running each function for different input values for the task which the function was created. For all situations where a function was not able to reach a conclusion, a new *if then* expression was added to the existing set.

The completeness checking of Xtimela-CBR components has been undertaken systematically during the implementation life cycle. As mentioned before the methods available for completeness checking are mostly logic-inspired and, hence, they are suitable for rule-based systems (Preece A., 1990). As shown above, some of these logic-based methods have been used to check the completeness of methods and functions represented in Xtimela-CBR. Other, more suitable methods have been used to check the completeness of the application case data base and domain models.

8.4.2- VALIDATION OF Xtimela-CBR

Validation of Xtimela-CBR was undertaken during all stages of its implementation cycle, which are: i) *the system skeleton*; ii) *the demo system*; and iii) *the working system.* The aims set at the beginning for validating Xtimela-CBR were:

- to determine the overall performance and effectiveness of the system in addressing the task of the assessment of grant applications with respect to the task structure outlined in chapter 4;
- to determine the utility and validity of the research hypothesis outlined in chapter 1; and
- to determine the Client's acceptance of the system.

To achieve the above aims, and taking into account the difficulties and advantages outlined in sections 8.3.1 and 8.3.2, the validation of Xtimela-CBR was based on the following requirements:

- the use of past successful grant applications to test the validity of the system against known results;
- the use of new applications to test the performance and effectiveness of the system in addressing the task;
- the use of Client's experts and external experts for testing the system with new grant applications; and
- the use of hypothetical test cases to reflect extreme conditions under which the system can be operated.

A description of the validation of the Xtimela-CBR using the methods listed in table 8.2 is given in the following sections.

8.4.2.1- Validation During System's Implementation

Validation during the systems implementation was the first step in the validation process. It was crucial to ensure that the following issues were covered at the skeleton stage:

- The system should retrieve only and all the appropriate application cases while processing the fewest possible probes, in order to achieve a good level of precision and recall.

- The validity and utility of the application cases stored in the case library.

If these issues were not assured, before any other validation tests, all of validation process would be jeopardised and the Client's interest on the system would not be ensured. For this reason, special attention was dedicated to those issues during the system implementation.

Simoudis (Simoudis E., 1992) employed three tests to validate the performance of CASCADE's prototype retrieval mechanism including:

1. The first test involved the top-down retrieval of cases using surface features, where each case was treated as a new problem.
2. The second test evaluated the effectiveness of the case retrieval.
3. The final test assessed the benefits of the surface-feature retrieval.

Hennessy and Hinkle (Hennessy D. and Hinkle D., 1992) found problems in validating the case memory of CLAVIER CBR system. To address this problem they decided to develop a validation tool which will automatically update the case memory and ensure that it is continuously validated.

Based on above experiences the validation during the system's implementation of both case library and case-base retrieval mechanisms consisted of:

1. Firstly, validating the 60 seed application cases stored in the case library, treating each one as a new application case. Thus, each of the seed application cases was validated one by one as a new problem and the results were compared to the information contained in each corresponding grant application.
2. Secondly, during the validation of the 60 seed application cases the performance of the case-base retriever component and indexing scheme was tested in terms of: i) the precision of the case retrieval; and ii) the percentage of recall of application cases.
3. Thirdly, activities 1 and 2 were repeated until 100% precision was found and 100% recall for the 60 seed application cases.
4. Fourthly, the activities mentioned in 1 and 2 were repeated for each case added to the case library either manually or by the system after each successful running.

This approach was revealed to be time consuming but effective. Initially, the performance was poor (100% precision for 25 of the 60 cases and 100% recall for 42 of 60 cases), but it improved over time until 100% precision was found and 100% recall was established for all of the 60 cases. Modifications were introduced as a consequence of errors detected in the process. Figure 8.1 illustrates the process followed.

8.4.2.2- Predictive Validation

Predictive validation was carried out after the validation of the initial 60 seed application cases and the case-base retriever, as described in section 8.4.2.2. The predictive validation of Xtimela-CBR consisted of:

- Comparing, in a very detailed way, solutions generated by the experts and other systems contained in each past grant application file with those provided by the system for the same grant application.
- Measuring the performance and functional features of the system.
- Validating against criteria.

This evaluation was mostly carried out by the author with assistance of client experts when major problems were found. Client's experts, during the predictive validation, provided useful suggestions to improve the performance of the system. Predictive validation was carried out together with refinement, so that after a detected problem the system was more valid and efficient.

Twenty five past grant applications were employed for predictive validation, all of them different from the 60 seed application cases. These 25 test grant applications were selected from a sample of 100 applications provided by the Client which were successfully processed and implemented. These 25 test cases were chosen because:

- The information contained in their descriptions (in terms of applicant and property) were enough complete for carrying out a meaningful analysis.
- They correspond to a fair variety of application types for renovation grants.

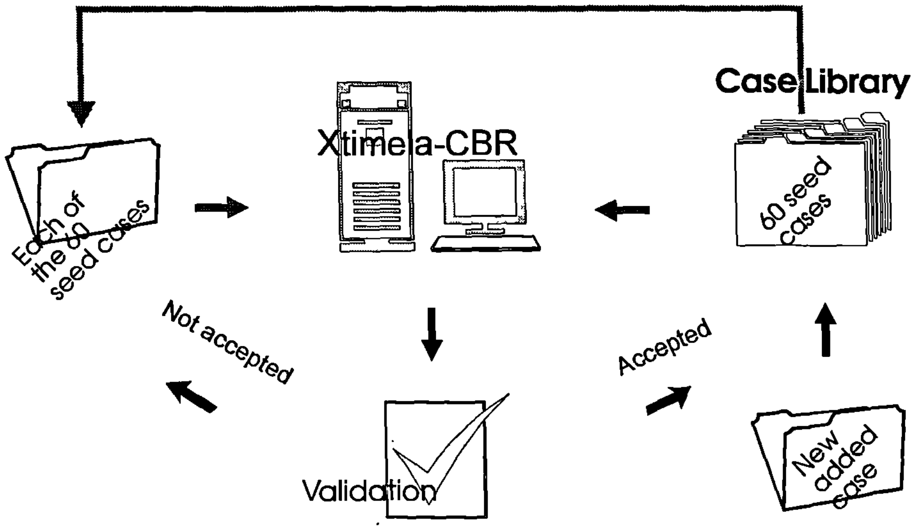


Figure 8.1: Illustration of the process followed for the validation of the seed cases

The description of the test cases is summarised in table 8.3.

Table 8.3: Summarised description of the test cases used for predictive validation

Application Code	Grant Type	Type Of Property	ondition O Property	Severity Of Unfitness	Type Of Applicant	Status Of Employment
15302351	Renovation	House	Unfit	2	Owner-occupier	Income support
1008527	Renovation	House	Unfit	3	Owner-occupier	Employed earner
18535553	Renovation	House	Unfit	3	Owner-occupier	Income support
6387317	Renovation	House	Unfit	5	Owner-occupier	Income support
14184168	Renovation	House	Unfit	3	Owner-occupier	Employed earner
13103949	Renovation	House	Unfit	5	Owner-occupier	Employed earner
174499276	Renovation	House	Unfit	4	Owner-occupier	Employed earner
13097338	Renovation	House	Unfit	2	Owner-occupier	Income support
6380799	Renovation	House	Unfit	3	Owner-occupier	Pensioner
163417	Renovation	House	Unfit	1	Owner-occupier	Pensioner
21011276	Renovation	House	Unfit	4	Owner-occupier	Employed earner
103257	Renovation	House	Unfit	4	Owner-occupier	Employed earner
14216552	Renovation	House	Unfit	2	Owner-occupier	Income support
13136911	Renovation	House	Unfit	2	Owner-occupier	Employed earner
13163922	Renovation	House	Unfit	2	Owner-occupier	Employed earner
1075116	Renovation	House	Unfit	2	Owner-occupier	Pensioner
53537	Renovation	House	Unfit	3	Owner-occupier	Income support
131405	Minor Works	House	Fit		Owner-occupier	
13144134	Disabled Facilities	House	Fit		Owner-occupier	Employed earner
2068312	Renovation	House	Unfit	2	Owner-occupier	Pensioner
532853	Renovation	House	Unfit	4	Owner-occupier	Employed earner
5343522	Renovation	House	Unfit	2	Owner-occupier	Employed earner
15287315	Renovation	House	Unfit	2	Owner-occupier	Employed earner
53429101	Disabled Facilities	House	Fit		Owner-occupier	Income support
15287315	Renovation	House	Unfit	3	Owner-occupier	Employed earner

Severity of unfitness means the number of requirements of the fitness standard which the property failed to pass.

To guide the predictive validation and assess results a standard form was created. This form which is presented in Appendix 9 was designed by the author specifically for this research project. The form groups the variables which were employed for carrying out the predictive validation into three main headings:

- Validation of intermediate and final results: groups the variables to validate the solutions generated by the system. It validates by comparing the solutions generated by the system with the ones provided by human experts.
- Validation by criteria: groups the variables to assess how the system addresses the tasks of the assessment of grant applications.
- Performance and functional features of the system: groups variables to assess the overall system performance and the validity of the research hypotheses.

For predictive validation it was necessary to use thirty nine variables, so that the reasons behind any discrepancies between the system's and the experts' results could be traced. Not all of the variables could be considered in every test case, because of the changes that occurred in the legal framework during the period of time covered by the test cases. The results of the predictive validation are presented in the following sections.

8.4.2.2.1- Enquiry Eligibility

The expert's decision on the eligibility of the enquiry for each test case was initially compared to the decision suggested by Xtimela-CBR. Also, the system's effectiveness and its accuracy in addressing the enquiry eligibility were evaluated. Table 8.4 summarises the results of the predictive validation for the enquiry eligibility task.

As shown by table 8.4 the system suggested the same decision as the one provided by the human experts for all test cases.

The validation by criteria consisted of evaluating the performance of the system with respect to three variables:

- Accuracy and correctness of the solution.
- Quality of the solution.
- Usefulness of the solution.

The objective of validation by criteria for this task was to assign a value to each variable mentioned above after each test case has been processed by the system.

Table 8.4: Comparison of enquiry eligibility task

Application	Expert's	System's		Criteria	
Code	**Decision**	**Decision**	**Accuracy**	**Quality of Solution ***	**Usefulness**
15302351	Proceed with application	Proceed with application	Equal to past result	Somewhat better than past result	Contains all useful information
1008527	Proceed with application	Proceed with application	Equal to past result	Somewhat better than past result	Contains all useful information
18535553	Proceed with application	Proceed with application	Equal to past result	Somewhat better than past result	Contains all useful information
6387317	Proceed with application	Proceed with application	Equal to past result	Somewhat better than past result	Contains all useful information
14184168	Proceed with application	Proceed with application	Equal to past result	Somewhat better than past result	Contains all useful information
13103949	Proceed with application	Proceed with application	Equal to past result	Somewhat better than past result	Contains all useful information
174499276	Proceed with application	Proceed with application	Equal to past result	Somewhat better than past result	Contains all useful information
13097338	Proceed with application	Proceed with application	Equal to past result	Somewhat better than past result	Contains all useful information
6380799	Proceed with application	Proceed with application	Equal to past result	Somewhat better than past result	Contains all useful information
163417	Proceed with application	Proceed with application	Equal to past result	Somewhat better than past result	Contains all useful information
21011276	Proceed with application	Proceed with application	Equal to past result	Somewhat better than past result	Contains all useful information
103257	Proceed with application	Proceed with application	Equal to past result	Somewhat better than past result	Contains all useful information
14216552	Proceed with application	Proceed with application	Equal to past result	Somewhat better than past result	Contains all useful information
13136911	Proceed with application	Proceed with application	Equal to past result	Somewhat better than past result	Contains all useful information
13163922	Proceed with application	Proceed with application	Equal to past result	Somewhat better than past result	Contains all useful information
1075116	Proceed with application	Proceed with application	Equal to past result	Somewhat better than past result	Contains all useful information
53537	Proceed with application	Proceed with application	Equal to past result	Somewhat better than past result	Contains all useful information
131405	Proceed with application	Proceed with application	Equal to past result	Somewhat better than past result	Contains all useful information
13144134	Proceed with application	Proceed with application	Equal to past result	Somewhat better than past result	Contains all useful information
2068312	Proceed with application	Proceed with application	Equal to past result	Somewhat better than past result	Contains all useful information
532853	Proceed with application	Proceed with application	Equal to past result	Somewhat better than past result	Contains all useful information
5343522	Proceed with application	Proceed with application	Equal to past result	Somewhat better than past result	Contains all useful information
15287315	Proceed with application	Proceed with application	Equal to past result	Somewhat better than past result	Contains all useful information
53429101	Proceed with application	Proceed with application	Equal to past result	Somewhat better than past result	Contains all useful information
19660526	Proceed with application	Proceed with application	Equal to past result	Somewhat better than past result	Contains all useful information
	* It was considered somewhat better than the experts decision because the system provides an explanation of how to achieve the decision.				

8.4.2.2.2- Fitness Assessment

The expert's decisions on the fitness of the property and the subsequent reasons for unfitness for each test case were initially compared to the decision and reasons suggested by Xtimela-CBR. Also, the system's effectiveness and its accuracy in addressing the fitness assessment task were evaluated. Table 8.5 summarises the results of the decisions about the fitness of each dwelling-house. As shown by table 8.5, the system suggested the same decision as the one provided by the human experts for the fitness of the property for all of the test cases. The validation by criteria consisted of evaluating the performance of the system with respect to three variables:

- Accuracy and correctness of the solution.
- Quality of the solution.
- Usefulness of the solution.

Table 8.5: Comparison of the decision about the fitness of the property

Application Code	Expert's Decision	System's Decision	Criteria: Accuracy	Criteria: Quality of Solution *	Criteria: Usefulness
15302351	Unfit for human habitation	Unfit for human habitation	Equal to past result	Equivalent to past result	Contains all useful information
1008527	Unfit for human habitation	Unfit for human habitation	Equal to past result	Equivalent to past result	Contains all useful information
18535553	Unfit for human habitation	Unfit for human habitation	Equal to past result	Equivalent to past result	Contains all useful information
6387317	Unfit for human habitation	Unfit for human habitation	Equal to past result	Equivalent to past result	Contains all useful information
14184168	Unfit for human habitation	Unfit for human habitation	Equal to past result	Equivalent to past result	Contains all useful information
13103949	Unfit for human habitation	Unfit for human habitation	Equal to past result	Equivalent to past result	Contains all useful information
174499276	Unfit for human habitation	Unfit for human habitation	Equal to past result	Equivalent to past result	Contains all useful information
13097338	Unfit for human habitation	Unfit for human habitation	Equal to past result	Equivalent to past result	Contains all useful information
6380799	Unfit for human habitation	Unfit for human habitation	Equal to past result	Equivalent to past result	Contains all useful information
163417	Unfit for human habitation	Unfit for human habitation	Equal to past result	Equivalent to past result	Contains all useful information
21011276	Unfit for human habitation	Unfit for human habitation	Equal to past result	Equivalent to past result	Contains all useful information
103257	Unfit for human habitation	Unfit for human habitation	Equal to past result	Equivalent to past result	Contains all useful information
14216552	Unfit for human habitation	Unfit for human habitation	Equal to past result	Equivalent to past result	Contains all useful information
13136911	Unfit for human habitation	Unfit for human habitation	Equal to past result	Equivalent to past result	Contains all useful information
13163922	Unfit for human habitation	Unfit for human habitation	Equal to past result	Equivalent to past result	Contains all useful information
1075116	Unfit for human habitation	Unfit for human habitation	Equal to past result	Equivalent to past result	Contains all useful information
53537	Unfit for human habitation	Unfit for human habitation	Equal to past result	Equivalent to past result	Contains all useful information
131405	*	*			
13144134	*	*			
2068312	Unfit for human habitation	Unfit for human habitation	Equal to past result	Equivalent to past result	Contains all useful information
532853	Unfit for human habitation	Unfit for human habitation	Equal to past result	Equivalent to past result	Contains all useful information
5343522	Unfit for human habitation	Unfit for human habitation	Equal to past result	Equivalent to past result	Contains all useful information
15287315	Unfit for human habitation	Unfit for human habitation	Equal to past result	Equivalent to past result	Contains all useful information
53429101	*	*			
19660526	Unfit for human habitation	Unfit for human habitation	Equal to past result	Equivalent to past result	Contains all useful information

* Not required to assess the fitness because of the type of grant sought

Table 8.6 summarises the results of the decisions about the reasons for unfitness and the severity of unfitness of each dwelling-house.

Table 8.6: Comparison of the decision about the reasons for unfitness and the severity of unfitness

Application	Reasons for unfitness		everity of Unfitness		Diff		Criteria	
Code	Expert's	System's	Expert's	System's	rence	Accur.	Quality	Useful.
15302351	Serious Disrepair & Prejudicial Dampness	Serious Disrepair & Prejudicial Dampness	2	2	0	Equal	Equivalent	Useful
1008527	S.Disrepair,P.Dampness&S.Unstable	S.Disrepair,P. Dampness&S.Unstable	3	3	0	Equal	Equivalent	Useful
18535553	Serious Disrepair & Prejudicial Dampness	S.Disrepair,P. Dampness&Inad.Lighting	2	3	1	Close	Equivalent	Useful
6387317	Not provided	.Disr.,P.Dampness,Ins.Cook.,Inad.Ligh.,Inad. Wa	-			-	-	Useful
14184168	S.Disrepair,P. Dampness&S.Unstable	S.Disrepair,P.Dampness&S.Unstable	3	3	0	Equal	Equivalent	Useful
13103949	.Disr.,P.Dampness,S. Unstable,Ins.Cook.,Inad.Lig	.Disr.,P.Dampness,S. Unstable,Ins.Cook.,Inad.Lig	5	5	0	Equal	Equivalent	Useful
174499276	S.Disr.,P.Dampness,S. Unstable&Inad.Lighting	S.Disr.,P.Dampness,S. Unstable&Inad.Lighting	4	4	0	Equal	Equivalent	Useful
13097338	Serious Disrepair & Prejudicial Dampness	Serious Disrepair & Prejudicial Dampness	2	2	0	Equal	Equivalent	Useful
6380799	S.Disrepair,P.Dampness & Inad.Lighting	S.Disrepair.,P.Dampness & Inad.Lighting	3	3	0	Equal	Equivalent	Useful
163417	Serious Disrepair	Serious Disrepair	1	1	0	Equal	Equivalent	Useful
21011276	Not provided	.Disrepair,P.Dampness,Ins.Cooking&Inad.Lightin	-	4	-	-	-	Useful
103257	Serious Disrepair	S.Disr.,P.Dampness,S. Unstable&Inad.Lighting	1	4	3	Far	better	Useful
14216552	Serious Disrepair & Structural Unstable	Serious Disrepair & Structural Unstable	2	2	0	Equal	Equivalent	Useful
13136911	Serious Disrepair & Prejudicial Dampness	Serious Disrepair & Prejudicial Dampness	2	2	0	Equal	Equivalent	Useful
13163922	Serious Disrepair & Prejudicial Dampness	Serious Disrepair & Prejudicial Dampness	2	2	0	Equal	Equivalent	Useful
1075116	Serious Disrepair & Structural Unstable	Serious Disrepair & Structural Unstable	2	2	0	Equal	Equivalent	Useful
53537	S.Disrepair,P. Dampness&S.Unstable	S.Disrepair,P. Dampness&S.Unstable	3	3	0	Equal	Equivalent	Useful
131405	*	*						Useful
13144134	*	*						Useful
2068312	Serious Disrepair & Prejudicial Dampness	Serious Disrepair & Prejudicial Dampness	2	2	0	Equal	Equivalent	Useful
532853	.Disrepair, P.Dampness,Ins.Cooking&Inad.Lighti	.Disrepair,P.Dampness,Ins.Cooking&Inad.Lightin	4	4	4	Equal	Equivalent	Useful
5343522	Serious Disrepair & Structural Unstable	Serious Disrepair & Structural Unstable	2	2	0	Equal	Equivalent	Useful
15287315	Serious Disrepair & Inadequate Lighting	Serious Disrepair & Inadequate Lighting	2	2	0	Equal	Equivalent	Useful
53429101	*	*						
19660526	S.Disrepair,P. Dampness&S.Unstable	S.Disrepair,P. Dampness&S.Unstable	3	3	0	Equal	Equivalent	Useful
	* Not required to assess the fitness because of the type of grant sought							

As shown by table 8.6 above, the system suggested the same decision as the one provided by the human experts for the reasons for unfitness for 90% of test cases. In one case the difference is due to the criteria used to define the failures belonging to the lighting requirement of the fitness standard. In others it might be caused by the test case having incomplete information. Both of these differences have been analysed and corrections were undertaken. The validation by criteria consisted of evaluating the performance of the system with respect to three variables:

- Accuracy and correctness of the solution.
- Quality of the solution.
- Usefulness of the solution.

8.4.2.2.3- Means Test

The experts' results (obtained with support of a computer software) about the means test of each test case were initially compared to the results suggested by Xtimela-CBR. Also, the system's effectiveness and its accuracy in addressing the means test task were evaluated. Table 8.7 summarises the results about the means test task.

At the time of their assessment, the means test for all twenty five test cases was assessed by human experts using computer software for that specific purpose. These means tests have been carried out according the regulations in force at the time of their assessment.

Xtimela-CBR was programmed to perform the means test according to the current regulations. Only nine of the tests cases were assessed according to the current regulations. Differences shown in the table 8.7 between the expert's and system's results in the *applicable amount*, *eligible income* and *reduction of grant* are mainly due to the differences in the regulations used and, in few cases, due to the Xtimela-CBR errors. For eight out of the nine test cases where the means test was carried out according to the same regulations the system, suggested the same results as the ones provided by the human expert. For the remaining one, the test case did not provide data about the means test. Because the goal of the means test is to find out the reduction in the amount of grant, the validation by criteria was carried out only with respect to this variable.

Table 8.7: Comparison of the means test results

Application Code	pplicable Amount		Eligible Income		Reduc. on Grant		Diffe-rences	Criteria		
	Expert's	System's	Expert's	System's	Expert's	System's		Accuracy	Quality	Useful.
15302351**	145.05	41	0	0	0	0	0	-	Equiva.	Useful
1008527*	155.85	155.85	225.4	225.4	1558.53	1558.53	0	Equal	Equiva.	Useful
18535553*	41	41	0	0	0	0	0	Equal	Equiva.	Useful
6387317**	75.4	41	0	0	0	0	0	-	Equiva.	Useful
14184168**	131.05	103.3	133.58	113.82	43.26	561.24	517.98	-	-	-
13103949**	142.8	141.75	168.15	184.55	1352.42	731.87	620.55	-	-	-
174499276*	68.5	88	221.68	222.17	7894.9	13715.4		-	-	Not U.
13097338**	93	41	0	0	0	0	0	-	Equiva.	Useful
6380799*	101.3	101.3	64.5	64.5	0	0	0	Equal	Equiva.	Useful
163417**	68.5	101.3	57.7	57.7	0	0	0	-	Equiva.	Useful
21011276**	142.7	142.7	212.78	212.79	1917.54	2071.32	153.46		S poor	Not U.
103257**	208.4	178	254.25	784.04	784.04	1101.67	317.63	-	-	Useful
14216552*	41	41	0	0	0	0	0	Equal	Equiva.	Useful
13136911**	99.4	126.75	153.63	138	2893.17	192.37		-	-	Not U.
13163922*	135.25	135.25	162.84	162.24	471.79	471.79	0	Equal	Equiva.	Useful
1075116**	75.4	101.3	63.28	63.28	0	0	0	-	Equiva.	Useful
53537**	56.7	41	0	0	0	0	0	-	Equiva.	Useful
131405***	-	-	-	-				-	-	-
13144134**	101.95	103.3	184.36	173.36	4247.41	1576.11	2671.3	-	-	-
2068312**	107.6	138	88.02	88.02	0	0	0	-	Equiva.	Useful
532853*	not avai.	101.3	not avai.	70.45	0	0	0	-	Equiva.	Useful
5343522**	109	77.6	188.7	207.87	5726.09	2424.67	3301.4	Equal	Equiva.	Useful
15287315**	105.55	133.7	218.94	250.88	6049.36	5371.97	677.39	-	-	Useful
53429101*	41	0	0	0	0	0	0	Equal	Equiva.	Useful
19660526*	41	0	0	0	0	0	0	Equal	Equiva.	Useful

* Test case was performed according to the new regulations

* * Test case was performed according to the old regulations

* * * The means test aws not necessary

test cases where the system failed to provide right solutions

8.4.2.2.4- Schedules Of Work

The experts' schedules of works to make the property fit were initially compared to the ones suggested by Xtimela-CBR for same property. Also, system effectiveness and its accuracy in addressing the schedule of work task were evaluated. Table 8.8 summarises the results of the schedules of work task.

As shown in the table 8.8, the system suggested the same schedules of work as the ones provided by the expert for all test cases. The validation by criteria consisted in evaluating the system performance with respect to three variables:

- Accuracy and correctness of the solution.
- Quality of the solution.
- Usefulness of the solution.

Table 8.8: Comparison of schedules of work results

Application Code	Expert's Results	System's Results	Criteria		
			Accuracy	Quality of Solution *	Usefulness
15302351	Same	Same	Equal to past result	Equivalent to past result	Reasonable contains all inf.
1008527	Same	Same	Equal to past result	Equivalent to past result	Reasonable contains all inf.
18535553	Same	Same	Equal to past result	Equivalent to past result	Reasonable contains all inf.
6387317	Same	Same	Equal to past result	Equivalent to past result	Reasonable contains all inf.
14184168	Same	Same	Equal to past result	Equivalent to past result	Reasonable contains all inf.
13103949	Same	Same	Equal to past result	Equivalent to past result	Reasonable contains all inf.
174499276	Same	Same	Equal to past result	Equivalent to past result	Reasonable contains all inf.
13097338	Same	Same	Equal to past result	Equivalent to past result	Reasonable contains all inf.
6380799	Same	Same	Equal to past result	Equivalent to past result	Reasonable contains all inf.
163417	Same	Same	Equal to past result	Equivalent to past result	Reasonable contains all inf.
21011276	Same	Same	Equal to past result	Equivalent to past result	Reasonable contains all inf.
103257	Same	Same	Equal to past result	Equivalent to past result	Reasonable contains all inf.
14216552	Same	Same	Equal to past result	Equivalent to past result	Reasonable contains all inf.
13136911	Same	Same	Equal to past result	Equivalent to past result	Reasonable contains all inf.
13163922	Same	Same	Equal to past result	Equivalent to past result	Reasonable contains all inf.
1075116	Same	Same	Equal to past result	Equivalent to past result	Reasonable contains all inf.
53537	Same	Same	Equal to past result	Equivalent to past result	Reasonable contains all inf.
131405	Same	Same	Equal to past result	Equivalent to past result	Reasonable contains all inf.
13144134	Same	Same	Equal to past result	Equivalent to past result	Reasonable contains all inf.
2068312	Same	Same	Equal to past result	Equivalent to past result	Reasonable contains all inf.
532853	Same	Same	Equal to past result	Equivalent to past result	Reasonable contains all inf.
5343522	Same	Same	Equal to past result	Equivalent to past result	Reasonable contains all inf.
15287315	Same	Same	Equal to past result	Equivalent to past result	Reasonable contains all inf.
53429101	Same	Same	Equal to past result	Equivalent to past result	Reasonable contains all inf.
19660526	Same	Same	Equal to past result	Equivalent to past result	Reasonable contains all inf.

8.4.2.2.5- Cost Of The Works, Economic Analysis And The Amount Of Grant

The experts' results about the cost of the works, economic analysis and the amount of grant were initially compared to the ones suggested by Xtimela-CBR for all test cases. Also, the system's effectiveness and its accuracy in addressing the cost of the works the economic analysis and the amount of grant were evaluated. Table 8.9 summarises the results.

As shown in the table 8.9 the system suggested the same cost for the schedules of work as the one provided by the expert for all test cases. The economic analysis is an innovation provided by Xtimela-CBR and in consequence no historical results were available to compare with. Economic analysis is recommended by the legal framework, but it is not currently carried out by the Client. In fact, the system confirmed what was assumed by the human experts regarding the economic merits of the renovation undertaken for all of the test cases.

Table 8.9: Comparison of the cost of works, the economic analysis and the amount of grant

Application	Cost Of Works		conomic Analysis		Amount Grant		Diffe-	Criteria		
Code	Expert's	System's	Expert's	System's	Expert's	System's	rences	Accuracy	Quality	Useful.
15302351**	3717.7	3717.7	Not prov.	Very Good	5640	4089.47	-1551	-	-	Useful
1008527*	9323.62	9323.62	Not prov.	Very Good	8541.1	8541.1	0	Equal	Equiva.	Useful
18535553*	18030.4	18030.38	Not prov.	Very Good	3648.28	3648.28	0	Equal	Equiva.	Useful
6387317**	16608.6	16608.62	Not prov.	Very Good	18560.1	18269.49	-290.6	-	-	Useful
14184168**	4615	4615	Not prov.	Very Good	7546.63	5917.3	-1629	-	-	-
13103949**	8394.2	8394.2	Not prov.	Very Good	8430	8428.57	-1.43	Close	Equiva.	Useful
174499276*	16508.8	16508.75	Not prov.	Very Good	3072.3	3070	-2.3	Close	Equiva.	Useful
13097338**	4529.62	4529.62	Not prov.	Very Good	5153.18	5529.62	376.44	-	-	Useful
6380799*	4299.32	4299.32	Not prov.	Very Good	4729.26	4729.26	0	Equal	Equiva.	Useful
163417**	4777	4777	Not prov.	Very Good	5612.97	6042.9	429.93	-	-	Useful
21011276**	9964	9964	Not prov.	Very Good	9372.71	9372.71	0	Equal	Equiva.	Useful
103257**	10892.3	10892.25	Not prov.	Very Good	11295.9	10336.65	-959.3	Far	S.poor	Not U.
14216552*	5540.12	5540.12	Not prov.	Very Good	6094.14	6094.14	0	Equal	Equiva.	Useful
13136911**	2179.62	2179.62	Not prov.	Very Good	0	2335.02	2335	Far	S.poor	Not U.
13163922*	838.36	838.36	Not prov.	Very Good	383.64	383.64	0	Equal	Equiva.	Useful
1075116**	6110	6110	Not prov.	Very Good	6721	6721	0	Equal	Equiva.	Useful
53537**	3901	3901	Not prov.	Very Good	429.1	390.1	-39	Close	Equiva.	Useful
131405***	861.35	861.35	-	-	880	880	0	Equal	Equiva.	Useful
13144134**	2230	2230	-	-	0	1148.56	1148.6	-	-	-
2068312**	2726	2726	Not prov.	Very Good	3046.31	3046.31	0	Equal	Equiva.	Useful
532853*	6092.38	6092.38	Not prov.	Very Good	Not prov.	6701.61	-	-	-	-
5343522**	2497.82	2497.82	Not prov.	Very Good	0	0	0	Equal	Equiva.	Useful
15287315**	6339.12	6339.12	Not prov.	Very Good	323.82	1063.68	739.86	-	-	Useful
53429101*	710	710	-	-	793	793	0	Equal	Equiva.	Useful
19660526*	88570.5	8570.45	Not prov.	Very Good	0	0	0	Equal	Equiva.	Useful

* Test case was performed according to the new regulations

** Test case was performed according to the old regulations

*** The means test was not necessary

Table 8.9 highlights the discrepancies in the amount of grant found respectively by the human expert and the system. For the majority of test cases it is a result of the differences in the regulations, as explained for the means test task. For other test cases it is a consequence of system's errors. For all applications processed according to the new regulations the system generated the same amount of grant as the one provided by the human expert. In only one case it was not possible to compare these details because historic data was not available in the test case.

8.4.2.2.6- Eligibility and Decision On Grant Application

The expert's decision was initially compared to the one suggested by Xtimela-CBR for each test case. The determination of the grant eligibility before a decision can be taken it is another innovation introduced by the system. It provides greater consistency in the decision. Also, the system effectiveness and its accuracy in addressing both tasks were evaluated. Table 8.10 summarises the results.

Table 8.10: Comparison of the application eligibility and grant decision results

Application	pplicatio Eligibility		Grant Decision		Overall	Criteria		
Code	Expert's	System's	Expert's	System's	Speed	Accuracy	Quality	Usefulness
15302351	-	High	Approve grant	Approve grant	25 minutes	Equal	Equiv.	Cont.all infor.
1008527	-	High	Approve grant	Approve grant	30 minutes	Equal	Equiv.	Cont.all infor.
18535553	-	High	Approve grant	Approve grant	30 minutes	Equal	Equiv.	Cont.all infor.
6387317	-	High	Approve grant	Approve grant	25 minutes	Equal	Equiv.	Cont.all infor.
14184168	-	High	Approve grant	Approve grant	30 minutes	Equal	Equiv.	Cont.all infor.
13103949	-	High	Approve grant	Approve grant	25 minutes	Equal	Equiv.	Cont.all infor.
174499276	-	High	Approve grant	Approve grant	35 minutes	Equal	Equiv.	Cont.all infor.
13097338	-	High	Approve grant	Approve grant	40 minutes	Equal	Equiv.	Cont.all infor.
6380799	-	High	Approve grant	Approve grant	35 minutes	Equal	Equiv.	Cont.all infor.
163417	-	High	Approve grant	Approve grant	30 minutes	Equal	Equiv.	Cont.all infor.
21011276	-	High	Approve grant	Approve grant	40 minutes	Equal	Equiv.	Cont.all infor.
103257	-	High	Approve grant	Approve grant	35 minutes	Equal	Equiv.	Cont.all infor.
14216552	-	High	Approve grant	Approve grant	30 minutes	Equal	Equiv.	Cont.all infor.
13136911	-	Very High	Approve grant	Approve grant	50 minutes	Equal	Equiv.	Cont.all infor.
13163922	-	Very High	Approve grant	Approve grant	35 minutes	Equal	Equiv.	Cont.all infor.
1075116	-	Very High	Approve grant	Approve grant	35 minutes	Equal	Equiv.	Cont.all infor.
53537	-	Very High	Approve grant	Approve grant	16 minutes	Equal	Equiv.	Cont.all infor.
131405	-	High	Approve grant	Approve grant	10 minutes	Equal	Equiv.	Cont.all infor.
13144134	-	Very High	Approve grant	Approve grant	30 minutes	Equal	Equiv.	Cont.all infor.
2068312	-	Very High	Approve grant	Approve grant	35 minutes	Equal	Equiv.	Cont.all infor.
532853	-	High	Approve grant	Approve grant	25 minutes	Equal	Equiv.	Cont.all infor.
5343522	-	High	Approve grant	Approve grant	30 minutes	Equal	Equiv.	Cont.all infor.
15287315	-	Very High	Approve grant	Approve grant	35 minutes	Equal	Equiv.	Cont.all infor.
53429101	-	Very High	Approve grant	Approve grant	35 minutes	Equal	Equiv.	Cont.all infor.
19660526	-	Very High	Approve grant	Approve grant	25 minutes	Equal	Equiv.	Cont.all infor.

As shown in table 8.10, the system suggested the same decision on the grant as the one provided by the human expert for all of the test cases.

The grant eligibility suggested by the system supports the decision achieved for each grant application. Table 8.10, also indicates the time taken to run the system and for it to perform all of the tasks of the assessment of applications for the HRGS. It was not possible to compare these numbers with the time taken by the human experts because of the multistage nature of the overall task. The validation by criteria consisted of evaluating the performance of the system with respect to three variables:

- Accuracy and correctness of the solution.
- Quality of the solution.

- Usefulness of the solution.

8.4.2.3- Field Validation

The predictive validation carried out with the knowledge-base refinement has demonstrated that Xtimela-CBR is suitable for field validation because it can be used experimentally in supporting the assessment of new grant applications without causing problems for the users. To guide the field validation, a standard form was designed and implemented for the field tests. This form which is presented in Appendix 9 embodies the variables used to evaluate the system in addressing the task for which it was built. The variables were grouped into two main headings: i) system's effectiveness and ii) user's effectiveness. Five different grant applications, were randomly selected for field validation. Their descriptions are summarised in Table 8.11.

Table 8.11: Summarised description of the grant applications employed for field validation

Application Code	Grant Type	Type Of Property	Condition Of Property	Severity Of Unfitnes	Type Of Applicant	Status Of Employment
174356154	Renovation	House	Unfit	6	Owner-occupier	Couple employed earner
64890785	Renovation	House	Unfit	3	Owner-occupier	Income support
21752810	Renovation	House	Unfit	4	Owner-occupier	Pensioner
64890785	Renovation	House	Unfit	3	Owner-occupier	Income support
174356154	Renovation	House	Unfit	3	Owner-occupier	Couple employed earner

Severity of unfitness means the number of requirements of fitness standard which the property failed to pass.

The Environmental Chief Officer of the Client was personally involved in the field tests. Due to his lack of familiarity with the system and interface features, the field tests were carried out with the support of the author. The outcome of the field tests regarding the overall effectiveness of the system are presented in table 8.12.

Table 8.12: Summarised description of the results from field validation

Applic. Code	Enquir Elig.	Fitness Asse.	Means Test	Sch.of Work	Econ. Anal.	Applic Elig.	Am. of Grant	Rep. Decis.	Ease of Input	Time to Run	Expla-nation
1.74E+08	Accep.	Accep.	Accep.	Accep.	Accep.	Accep.	Accep.	Accep.	R.accep.	45 min.	R.accep.
64890785	Accep.	Accep.	Accep.	Accep.	Accep.	Accep.	Accep.	Accep.	R.accep.	20 min	R.accep.
21752810	Accep.	Accep.	Accep.	Accep.	Accep.	Accep.	Accep.	Accep.	R.accep.	35 min	R.accep.
64890785	Accep.	Accep.	Accep.	Accep.	Accep.	Accep.	Accep.	Accep.	R.accep.	40 min	R.accep.
1.74E+08	Accep.	Accep.	Accep.	Accep.	Accep.	Accep.	Accep.	Accep.	R.accep.	30 min	R.accep.

Accep.: Acceptable
R.accep.: Reasonably acceptable

At the end of the field test the Client's Environmental Chief Officer answered a questionnaire in which he provided a very encouraging response. This questionnaire is presented in Appendix 9. Table 8.13 shows a summary of the answers given by the expert.

Table 8.13: Summary of Comments provided by the Environmental Chief Officer

Issues	Comments
Performance of the system	Satisfied
Suitability of the system in addressing the task	Suitable
Where the system is potentially useful	Fitness Assessment Disabled Facilities Assessment Schedule of Works Cost of Works Means Test
Benefits of the system	Time and Cost Savings Decision improvement Efficiency Improvement Consistency Accessibility to Expertise
Overall efficiency	Reasonably satisfied
Suitability of case-based reasoning model	Suitable
Appropriateness of object oriented language	Appropriate
Appropriateness of the indexing scheme used for the cases	Appropriate
Benefits from the economic analysis and degree of eligibility tasks	Some
User interface	Appropriate
Usefulness of explanations	Useful
Organisational learning benefits	Some
CBR allows the system to derive solutions quickly	Much
CBR allows the system to learn from experience	Much

Client's experts pointed out problems and suggested improvements to the system, which were introduced later. These problems were related to: i) the user interface; and ii) the use of redundant conditions for evaluating the eligibility of the enquiry. A modification to how the assessment of the fitness of the property is carried out was proposed by Client's experts. Instead of prompting the user with questions, they suggested to use a menu of evidences of failure.

8.4.2.4- Face Validation

The benefit of involving human experts in the validation process who were not involved in the system development has been highlighted by a number of authors (e.g., Brandon P., et al. 1988; and Tuthil S., 1990).

Despite the difficulty of involving experts not linked to the Client to undertake a face validation, three experts in information technology were invited to give an independent judgement on the performance of the system and particularly on the model employed. One of the experts declined the invitation due to lack of his time. The other two experts were generally impressed with the system, and they gave very encouraging opinions. Face validation was carried out into two phases by using a questionnaire designed by the author to guide the tests:

- The first phase, involved a briefing about the system and a detailed demonstration of its performance.
- The second phase, consisted of running the system using several application grants in order to allow the invited experts to test the performance of the system in detail.

During the second phase, the experts completed the questionnaire mentioned above. These completed questionnaires are presented in Appendix 9. A summary of comments provided by the experts is presented in the table 8.14.

In broad terms, both experts agreed with the model employed to implement the system. They highlighted the potential benefits of CBR in the application area. As in the field validation, both experts also suggested improvements to the system's functionality, particularly related to the man-machine interface. These suggestions were implemented later. Problems pointed out by both experts were related with: i) the user interface; and ii) the lack of help and printing facilities.

8.4.2.5- Robustness Test

The robustness test was undertaken to test the behaviour of the system under extreme grant application conditions, regarding the households and dwelling-houses. Only the most important components of the system were tested. These components included: i) the application case database (case library and access procedures); ii) the new case indexer and iii) the case-base reasoner. These components were selected because much of the system's performance depends upon them. All robustness tests were carried by the author. A set of hypothetical grant applications was created, each case focusing on testing a particular aspect of the above mentioned system's components. The extreme conditions considered in the robustness test included:

- A set of six houses with extreme conditions of unfitness and severity of unfitness varying from 6 to 11;

- Three different types of applicants: i) a young couple of employed earners and having capital; ii) a couple where one member is disabled and the other is a pensioner; iii) couple with children and living on earnings from capital.

Table 8.14: Summary of Comments provided by the two invited experts

Issues	Expert A Comments	Expert B Comments
Performance of the system	Satisfied	Satisfied
Suitability of the system in addressing the task	Very Suitable	Suitable
Where the system is potentially useful	Preliminary enquiry Fitness Assessment Disabled Facilities Assessment Economic analysis Cost of Works Means Test Degree of Eligibility Final Decision	
Benefits of the system	Time and Cost Savings Decision improvement Efficiency Improvement Consistency Accessibility to Expertise Training Reporting Understand ability of domain	
Overall efficiency	Very satisfied	Very satisfied
Suitability of case-based reasoning model	Very suitable	Suitable
Appropriateness of object oriented language	Very Appropriate	Appropriate
Appropriateness of the indexing scheme used for the cases	Appropriate	
Appropriateness of retrieval mechanism	Appropriate	
Benefits from the economic analysis and degree of eligibility tasks	Very much	
User interface	Generally appropriate	Less appropriate
Usefulness of explanations	Neutral	Somewhat useful
Organisational learning benefits	Very much	Much
Case-based reasoning allows the system to derive solutions quickly	Very much	
Case-based reasoning allows the system to learn from experience	Very much	

In practice the tests undertaken were more concerned in validating the robustness of the case library, the retrieval and adaptation procedures encoded in the system. After each robustness test, refinement of the knowledge base was undertaken. The robustness test allowed to wider the range of experiences stored in the system. In this sense, the tests were useful for the improvement of the system.

8.5- SUMMARY AND CONCLUSIONS

This chapter introduced some fundamental concepts and methods related to the validation of KBSs. It also described the V&V of the system developed in this research. The V&V of Xtimela-CBR was an essential activity within system implementation cycle. It was carried out with certain pragmatism essentially due to existing constraints. These constraints were mostly concerned with the limited resources available for this research and the complexity of the task.

Experts from the Client and from other sources were involved in the validation. They provided very useful comments towards the improvement of Xtimela-CBR.

A total of nine different methods were employed: i) three for the verification; and ii) six for the validation. The main conclusions that resulted from the application of those nine methods are summarised as follows:

- One major advantage of integrating the V&V with the system implementation life cycle was that after each phase the system was more valid, correct, complete and consistent. This is because after each evaluation cycle, a refinement of the knowledge-base followed.
- The verification of the system carried out with its implementation was revealed to be a good programming practice. It saved time and costs and facilitated the validation process.
- Checking the complcteness of the case library has to be carried out systematically. Much of the performance of the system depended on the completeness of the case library.
- The validation of the initial case library during implementation was fundamental for the success of the following phases. The technique employed to validate the seed cases was revealed to be efficient and practical. With the initial case library validated it was possible to speed up the process of case acquisition.
- It was advantageous to employ several methods for the V&V of Xtimela-CBR. The experience has shown that they complement each other, because it was possible to focus on different aspects of the system's validity.
- The predictive validation was the most detailed validation carried out. Among other benefits, the predictive validation allowed the author to amplify the case library with additional representative application cases and to achieve substantial improvements in the system's functionality. Carrying out the predictive validation with an initial case library previously validated, proved to be an appropriate procedure for CBR systems.

Therefore, carrying out case acquisition with predictive validation fulfilled two roles: i) facilitated the case acquisition; and ii) validated every case added to the case library.

- Field validation was useful for capturing the Client's views and their acceptance of the system. It allowed the author to demonstrate the system's features and to highlight the potential benefits offered by the adopted model. Although there were constraints related to the system's functionality, it was worthwhile carrying out the field validation with the involvement of the Client. The Client's views were highlighted by the field tests and allowed the author to introduce improvements in the system. Field validation should be integrated with future developments of the system.
- Face validation allowed the author to capture the views of independent experts on the aspects of KBSs. It was useful for validatiing the model adopted to implement the system.
- A number of improvements introduced in the system were suggested by both the Client and independent experts. There are other suggestions for improvements which will be discussed in the next chapter.

Whether or not the system has reached an acceptable level of performance in general terms is subjective.

The V&V of Xtimela-CBR allowed the author to focus on evaluating the utility and validity of the research hypotheses outlined in chapter 1. The conclusions about the validity of the research hypotheses are presented in next Chapter 9.

CHAPTER 9

CONCLUSIONS AND FUTURE DIRECTIONS

The general objective of the present research project was to investigate the validity and suitability of CBR for implementing a KBS which can support human experts in assessing applications for the HRGS. This general objective was accomplished with the development of the Xtimela-CBR system described in the previous chapters. This chapter summarises the potential benefits exhibited by the system, the conclusions about the validity of the hypotheses presented in chapter 1, the contributions of the present research and the suggestions for future work. It concludes this thesis.

9.1- BENEFITS AND LIMITATIONS OF Xtimela-CBR

9.1.1- BENEFITS

Xtimela-CBR was conceived as a computer application with the main role of supporting Client's experts in assessing applications for the HRGS. Its main task associated with this role is to perform the different tasks of the assessment of grant applications for the HRGS.

The V&V of the system has highlighted the main features of the Xtimela-CBR and has suggested some benefits deriving from the integrative model adopted in this research. Basden (Basden A., 1994) proposed a taxonomy of benefits that he claims has the potential for the evaluation of the usefulness of a KBS in its working context. Basden's theory suggests that there are three levels of benefit to evaluate the effects and usefulness of KBSs. These three levels are:

1. Feature benefits: that arise from technological features of the system.
2. Task benefits: that arise from using the system in performing the task for which it was built.
3. Role benefits: that arise from the effect the system has on the roles the user fulfils by carrying out supported tasks.

The three levels benefit model from Basden is shown in figure 9.1.

The Basden's model structures the DTI's (Department of Trade and Industry, 1992) list of benefits of KBSs into the above mentioned three levels. Table 9.1 shows a partial view of the classification of the DTI's benefit list into the Basden's model.

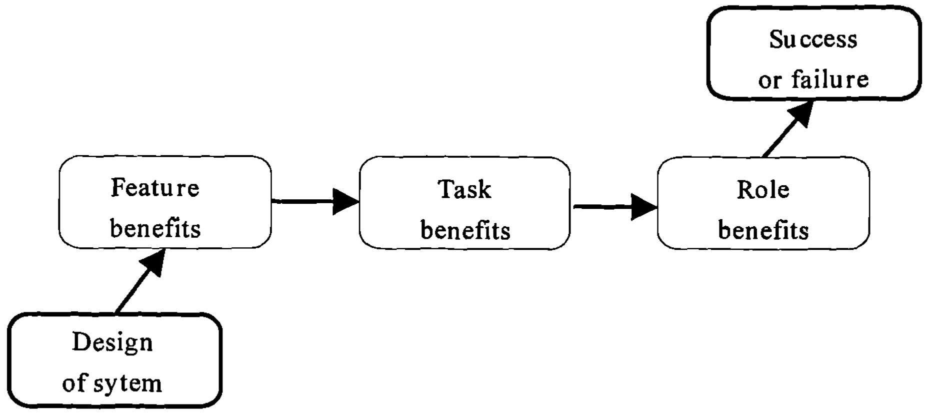


Figure 9.1: The three levels benefit model from Basden

The study carried out for evaluating the ELSIE KBS in use, suggested that there were three levels of benefits (Brandon P., 1993b):

- those that accrue from the technological features of the system;
- task level benefits which enhance the performance of individual tasks; and
- role level benefits which lead to changes in the surveyor's relationship with their customers.

This study highlights the major benefits arising from the use of the ELSIE system by different professionals working in the construction industry.

Following Basden's model and the ELSIE study, the valued characteristics arising from the development of Xtimela-CBR system were classified as: i) benefits from technological features of the system; ii) potential task benefits to users of the system; and iii) potential benefits at role level. The description of valued benefits of the Xtimela-CBR system is presented in the following sections.

9.1.1.1- Benefits From The Features Of The System

According to Sharma (Sharma R.S., 1992) technology refers both to the KBS per se and to its operation. For this author, the dimension of quality that falls under the technological features category essentially reflects the architectural and design features of the inference mechanisms, knowledge-base, explanation facility and user interface.

Table 9.1: DTI's benefits classified according Basden (from Basden A., 1994)

Role benefits	Task benefits	Feature benefits
Improved quality	Time savings	Consistency
Improved administration	Cost savings	Enhanced information flow
Reduced staff quantity	Improved response times	Better integration with systems
Reduced financial losses	Enhanced equipment usage	
Increased revenue	Involv. of staff of lower expert.	
Better customer services	More products offered to customers	
Better management decisions	Improved training	
Increased ability to compete	Easier maintenance	
More effective marketing/sales	Increas. availa. of exp. knowle.	
	Better unders. of problems	
	Increased effect.of managers	
	Additional training tool	
	Better understanding of techn.	
	Faster management decisions	

There were three main feature benefits identified during the implementation, validation and refinement of the system. These are as follows:

- **Learning:** In every circumstance in which a schedule of work is adapted by the case-based reasoner and the current application case stored and indexed in the case library, a new piece of knowledge is learned. This new knowledge can be useful in similar situations in future. Hennessy and Hinkle (Hennessy D. and Hinkle D., 1992) emphasised that CBR's most important benefit is its inherent ability to learn.
- **Increased accessibility to Client's expertise:** Xtimela-CBR stores and makes available Client's expertise for those using the system. This is because: i) the system's knowledge base records knowledge about how human experts have applied their experience and expertise in solving specific problems regarding households and property; and ii) the system is able to bring about all of and only of the knowledge which is required for the current task. Such ability has been emphasised by Client's experts.
- **Consistency and accuracy of decisions:** For some of the assessment tasks for HRGS applications, experience and judgement play a key role in reaching a decision. These tasks include: i) assessing the fitness of a dwelling-house; ii) finding out the schedules of work for a dwelling-house deemed unfit; and iii) assessing the merits of a given grant application. Human experts with different levels of experience can introduce variability in the decisions they produce. Therefore, LAs should consider each application for its own merits in reaching their decisions. Xtimela-CBR can provide more consistent and accurate results (in situations which it is familiar with) than that of an ordinary team of environmental experts for the following reasons: i) the system uses the knowledge gained from past experience to provide solutions for new problems; and ii) the system provides task-specific explanations for its suggestions. In so doing, the system has the ability: i) to maintain the consistency of the solutions it provides, in the context of the

current legal framework; and ii) to reduce the human expert variability by suggesting solutions which have proved useful in previous similar situations.

9.1.1.2- Task Benefits Resulting From The System

According to Brandon (Brandon P., 1993b) the task benefits to the users of a KBS are those which enhance the performance of individual tasks. Following this definition, the task benefits of Xtimela-CBR were evaluated regarding two distinct facets: i) the enhanced performance of the system in carrying out the overall task and its sub tasks as described by the task structure; and ii) the potential resulting changes to decision-making brought about by using the system. The benefits to users at task level included the following:

- **Improved response times:** Some of the tasks in the assessment of applications for the HRGS can involve lengthy processes. These tasks include: i) the assessment of the fitness of a dwelling-house; ii) finding out the schedules of work; iii) finding out the economic merits of the renovation action; and iv) means testing. Performing these tasks involves: i) accessing a variety of information sources; ii) processing large amounts of data; iii) complex computations; and iv) significant judgements based on experience. During the validation and refinement, Xtimela-CBR has demonstrated to be able to derive solutions quickly for these tasks. This was possible because: i) the system combines several problem solving methods to achieve solutions for the different assessment tasks; and ii) an important part of the system's knowledge comes from past experience. These characteristics of Xtimela-CBR speed up the assessment of a grant application and improve the response time of the experts to the applicants.
- **Reduction of the expert's cognitive work:** Xtimela-CBR provides support to human experts in assessing applications for the HRGS. It also provides the user with several types of knowledge which have been successfully used in previous situations by different experts. Therefore, this knowledge can be augmented by the experiences of those who use the system.
- **Easier maintenance:** One of the advantages of Xtimela-CBR is that the knowledge it contains can be easily maintained without having to continually reorganise the knowledge base when it is updated. Maintenance of the system is made easier because: i) the system can automatically add and index into its case library a current application case which has been successfully processed; and ii) the object-oriented approach adopted for the other components of the knowledge-base facilitates the updating of the system's main building blocks, such as objects and functions.

- **Provision of standard reports:** Xtimela-CBR provides intermediate task-specific reports of its results during each problem-solving session for assessing a grant application. At the end of the session, the system provides a final report of the solution plan found for the current application. The intermediate and final reports are organised in a standard format which displays: i) only and all information related with the current task (intermediate reports); and ii) all information contained in the solution plan for the current application. The reports provided by Xtimela-CBR include: i) the report on the enquiry eligibility; ii) the report on the assessment of the fitness of a dwelling-house; iii) the report on the assessment of disabled facilities; iv) the report on the budget; v) the report on the economic analysis of the renovation works; vi) the report on the test of resources of the persons relevant to the application; vii) the report on the degree of eligibility of the application; and viii) the solution plan for the current application.

9.1.1.3- Role Benefits Resulting From Developing The System

The benefits described above can lead to significant improvements in the relationship between the Client and the applicants because of the change in the Client's role. The improved efficiency brought about by using the system provides advantages both to the Client and the applicants. In particular, the increased speed at which the assessment tasks of grant applications are performed, resulting in organisational benefits, provides advantages to the Client. In consequence, the Client may be able to commit more time for dealing with strategic decisions of the HRGS. Therefore, the assistance to the applicants during the process can be improved by using the system and, hence, the relationship between the Client and the applicant can be enhanced.

9.1.2- LIMITATIONS OF THE SYSTEM

Although there are a number of benefits arising from the development of Xtimela-CBR, the following limitations of the system were found to exist:

- **Limitations on help and printing facilities:** Because of the limitations and aims of this research, Xtimela-CBR at its present stage does not provide help and printing facilities. The architecture discussed in chapter 6 was conceived to provide help to the users throughout all of the consultation processes. Printing reports are therefore essential for enhancing the role of the system.

- **Static screens:** The solutions displayed by Xtimela-CBR screens can not be modified by the user due to the user interface limitations. Dynamic screens are essential for enhancing the ability of the user to navigate through the reports provided by the system.
- **Limitation on domain problems:** Xtimela-CBR only covers three types of grants. Xtimela-CBR architecture can easily accommodate other domain problems. It does not cover applicants who are self-employed earners or students.

9.2- SUMMARY OF CONCLUSIONS

9.2.1- CONCLUSIONS

This research has successfully demonstrated that a KBS architecture which unifies different problem solving methods and knowledge types can be used for performing all tasks of the assessment applications for the HRGS. The general response given by the experts involved in the process of V&V of the system was good. These experts highlighted the potential benefits of the system and provided useful suggestions for later improvements. Some of these suggestions were related with the system's user interface.

Xtimela-CBR was implemented up to the stage of a working system. The present version includes: 155 real application cases; 741 objects; and 67 functions. As far as the literature in the field of construction is concerned, Xtimela-CBR is the first application of this kind, designed and implemented specifically for supporting human experts in assessing applications for the HRGS.

Xtimela-CBR has proved to perform as accurately as human experts do for all of the tasks for the assessment of grant applications. Therefore, the system has the potential to enhance the human response in crucial tasks such as:

- evaluation of the eligibility of an enquiry for a grant application;
- assessment of the fitness of a dwelling-house with respect to the fitness standard for human habitation;
- specification of the schedules of work to make fit a dwelling-house deemed unfit for human habitation;
- assessment of the adaptations necessary with respect to the disabled facilities recommended by the welfare authority for a disabled occupant;
- specification of the adaptation works required by a disabled occupant;
- specification of the minor works necessary to repair and improve a dwelling-house occupied by elderly occupants;

- economic evaluation of a given renovation action;
- means testing of the relevant persons for a given application for renovation or disabled facilities grants;
- determination of the amount of grant to be awarded; and
- finding a solution for applications in the following domains: renovation grants; disabled facilities grants; and minor works assistance.

At the end of assessing a current application case, the system provides the user with a solution plan presented in a format containing enough information for: i) awarding a grant to the applicant(s) if eligible; and ii) implementing the necessary renovations, disabled facilities or minor works. The validation of the system has demonstrated that the solution plans suggested by Xtimela-CBR are sufficiently accurate. The reasonably good content of the solutions and the increased speed at which the solutions are provided by the system were stressed by the Client as its strengths.

Xtimela-CBR introduces two innovations to the current process of assessing applications for the HRGS. Thus, Xtimela-CBR enables the Client's staff to perform:

- The economic analysis of the renovation works for an unfit dwelling-house.
- The determination of the degree of eligibility of the current application.

Both tasks are considered to be innovations because they are not currently performed by the Client, although they are recommended by the present legal framework. They can improve the quality of the solution plan sought for a given grant application. The economic analysis implements the economic formula prescribed by the legal framework for assessing the cost and benefits of the available alternatives for dealing with an unfit dwelling-house. The degree of eligibility of a grant application is determined by assessing the merits of the current application using a specific criteria. This criteria is based on those relevant aspects to which the legal framework draws attention.

The development of Xtimela-CBR has demonstrated that CBR is not fully suitable for the assessment of applications for the HRGS. Other techniques in combination with CBR were used to enable the development of Xtimela-CBR with the capability of performing all tasks of the assessment of applications for the HRGS. Xtimela-CBR unifies several methods within a single framework to support human experts in performing the assessment of grant applications. Therefore, applications cases stored in the case library do not support all tasks of the assessment of applications for the HRGS. To achieve the goal of the assessment of grant applications other types of knowledge were collected and represented in Xtimela-CBR knowledge base. Thus, the development of Xtimela-CBR does not support the *first and third hypothesis of this research.*

The vocabulary established for representing and indexing new and existing application cases in the Xtimela-CBR case library proved to be sufficiently stable and expressive for supporting the following reasoning tasks: i) finding the schedules of work required to make fit a dwelling-house deemed unfit for human habitation; ii) finding the schedules of adaptation works required for the disabled facilities recommended by the welfare authority; iii) finding the schedule of minor works; and i) finding a decision for the current application. The indexing vocabulary is a subset of the vocabulary used for the case representations. The indices have been demonstrated to be predictive, abstract, concrete and useful enough for guiding the search and matching in the case library. The system has proved to perform very well in terms of speed, accuracy and precision in deriving (for the three versions of the domain) schedules of work by retrieving and/or adapting old solutions contained in the application cases stored in its case library, thus supporting the *second hypothesis of this research.*

Xtimela-CBR automatically stores and indexes all application cases which it has successfully processed. As each application case is solved, it is added and indexed so that it can be used in the future to support those tasks where CBR is invoked. The validation has proved that in every circumstance in which a schedule of work is adapted by the case-based reasoner and the current application case stored and indexed in the case library, a new piece of knowledge is learned. This new knowledge can be useful in similar situations in future. However, the overall system capability and competence does not automatically fully increase over time by adding new application cases into the case library. This is because Xtimela-CBR uses other types of knowledge in addition to the application cases in supporting the different tasks of the assessment of grant applications. Therefore, only the case library is automatically up-date. These conclusions indicate that the *fourth hypothesis of this research* has not been fully proved.

As consequence of developing Xtimela-CBR, this research contributes in several areas such as:

- It provides a computer model which: i) integrates all of the tasks for the assessment of grant applications into a single system; and ii) unifies several problem solving methods and knowledge types to solve the overall task.
- It enhances the accuracy of the assessment of the fitness of a given dwelling-house.
- It enhances the accuracy and speed of deriving schedules of work for: i) making fit for human habitation a dwelling-house; ii) carrying out the adaptations required by a disabled person; and iii) carrying out minor works required by elderly people.
- It enables users to carry out the analysis of the economic merits of a proposed renovation action for a given dwelling-house.

- It provides a criteria for evaluating the merits of a given grant application.
- It provides a model for adapting a schedule of work using old solutions provided by useful application cases.

The task analysis, the implementation and validation of the system following the CCA method, were the main strengths of this research. The task structure provided a good specification for implementing and evaluating the system. The implementation of the system following the CCA method enabled the author to maintain the interest of the Client throughout the research project and obtain the support needed. The resulting system has successfully demonstrated how an integrative architecture and a case library can be used for supporting the assessment of applications for the HRGS. However, much work remains in order to obtain a finished application.

9.2.2- LIMITATIONS OF THIS RESEARCH

Although the aims of this research have been successfully attained, there are limitations. These include:

- The system's case library it is not complete in all of the three categories of application cases. Application cases for discretionary grants, both for renovation and disabled facilities grants are not available in the case library. In addition, the application cases for minor works only cover one of the four types of assistance currently available. This is because the Client currently does not process discretionary grant applications (renovation and disabled facilities) and other types of minor works grant applications. Thus, the system can not assess those discretionary grants. One solution to this problem is to include a number of prototypes of such application cases in the system. These prototypes should be abstract enough to represent a wide range of situations and concrete enough to enable the system to use them for problem solving. They should have representation structures similar to the ones used by the application cases currently stored in the case library.
- The knowledge represented in the system's domain models and procedures for performing the assessment of fitness of a dwelling-house is sufficiently representative and covers the current needs of applications processed by the Client. However, the system has no way to deal with situations which go beyond its own knowledge about how to derive of grounds of unfitness based on expert's observations. Adding new knowledge to the domain models has to be done manually by someone with working knowledge of the programme. However, the system's modularity provided by the task structure makes it easy to maintain the domain models.

- As each new application case is solved, it can be added to the case library. This leads to a rapid increase in the number cases in the case library over time, considering the current pace of applications received by the Client. To overcome this problem, the system's case-base reasoner was programmed to stop the search when it finds the first most similar application case. However, it therefore discards other potential matching cases which might provide a better solution. On the other hand, the system can not distinguish those cases which add new knowledge from those which do not add new knowledge when it is storing and indexing application cases. Solutions to this problem include: i) enabling the system to delete those application cases which do not add any new knowledge or which are no longer useful; or ii) enabling the system to distinguish cases which add new knowledge from those which do not and to store them in different partitions of the case library. Those cases which add knowledge to the system can be stored in a partition which supports problem solving. Those cases which do not add new knowledge can be stored in a partition where they can be accessed by the user for browsing purposes.

9.3- LESSONS FOR THE FUTURE

Some of the most important contributions of this research are the lessons learned from the development of Xtimela-CBR. It is hoped that these lessons will be valuable in the future. They fall into the following categories: i) task analysis; ii) acquisition of application cases; iii) case representation and indexing; iv) case library organisation; v) case retrieval and adaptation; vi) implementation of the system; and vii) verification and validation of the system. The following sections summarise the lessons for the future presented in accordance with the above categories.

9.3.1- TASK ANALYSIS

The task analysis discussed in chapter 4 provided a description of the proposed system in terms of the knowledge it contains: i) tasks and goals; ii) problem solving methods as ways to accomplish the tasks; iii) sub-tasks set by the problem solving methods; iv) sub task proposal knowledge; and v) search and control knowledge. The three versions of the task structure developed as a consequence of the task analysis provided a good specification of the system which has been useful for: i) designing the system's architecture discussed in chapter 6; ii) implementing and validating the system; and iii) guiding the acquisition of the task knowledge discussed in chapter 5. The task structure gives modularity to the system by organising sub-tasks and knowledge into problem spaces. This enables the system to use all of and only the knowledge required to solve the current task.

9.3.2- ACQUISITION OF APPLICATION CASES

Although an important part of the effort committed to knowledge acquisition was for the purpose of case acquisition, it has nevertheless proved fruitful. This was because acquiring the specific knowledge held in past grant applications in terms of cases was revealed to be a much easier and faster process than the acquisition of other types of knowledge, such as procedures and domain models. On the other hand, acquiring knowledge from human experts using past grant applications was also a very useful procedure because: i) it enabled more efficient use of the limited time available from the human experts; and ii) human experts were more motivated to provide their knowledge in terms of their specific experiences from past assessments. The general principles adopted to guide the case acquisition (discussed in chapter 5) have been revealed to be very useful for knowledge acquisition in general. It was an incremental process which enabled the author to build a case library to support the reasoning tasks that CBR is responsible for. The acquisition and validation of the initial 60 seed application cases played a key role in the implementation of the system. Validating and refining the content and representational structure of each application case added to the case library, both manually or by using the system, was a very useful procedure because it enabled the author to work with an updated case library during all of the stages of the system's implementation. Therefore, it allowed the author to keep track of: i) the experiences provided by each application case; ii) the system's behaviour; and iii) the range of problems found in the domain.

9.3.3- CASE REPRESENTATION AND INDEXING

The case acquisition process, followed by validation and refinement, provided a vocabulary for representing and indexing new and existing application cases. Much of the performance of the case-base reasoner relies upon the way in which: i) application cases are represented and indexed in the case library; and ii) application cases are accessed in the case library. The process followed for developing the three categories of case representations and the six types of indexing schemes has worked well, in the light of the results obtained through the system's validation for the tasks where CBR is applied.

9.3.4- CASE LIBRARY ORGANISATION

The hierarchical organisation of the case library enables the system to store and to partially index past application cases. This hierarchical organisation partitions the case library so that only a relevant portion of it is accessed by the retrieval procedures. Such organisation has the advantage of making the search and matching in the case library more efficient. The Xtimela-CBR case library provides a way of visualising the relationships between several partially similar application cases, i.e., application cases which share some feature-values in common. The internal nodes of the case library organisation store generalised knowledge associated with partitions of partially similar cases. This knowledge is represented as objects associated to each node. These objects consist of the features and methods. As the case library grows in the number and range of problems, further nodes may be necessary to maintain an acceptable level of efficiency for the search. These nodes add more generalisations to the hierarchy. This general knowledge was revealed to be useful for: i) partial indexing of the application cases; ii) guiding the search; and iii) guiding the adaptation. Maintaining or expanding the case library is simply a process of: i) adding new application cases or deleting existing ones; ii) adding or deleting nodes of the hierarchy; and iii) adding, modifying or deleting case representation features. The inheritance and object network features provided by the Kappa-PC shell facilitated the implementation and refinement of the case library.

9.3.5- APPLICATION CASE RETRIEVAL AND ADAPTATION

Application case retrieval in Xtimela-CBR integrates search and matching functions for selecting a similar application case. Firstly, the case-base reasoner searches in the case library for a partition of partially matching application cases. Secondly, the case-base reasoner searches that partition for a full matching application case. If the system succeeds, it returns the similar application case. When performing the task of finding the schedules of work, if the case-base reasoner fails to select a full matching application case, then it backtracks and performs the adaptation of a schedule of work. This research provided a model for adaptation based on the principle that: i) a schedule of work can be decomposed into a set of primitive components; ii) some of these primitive components are provided by the ground of unfitness; iii) for the same ground of unfitness, different types of work components suggest different specifications of schedules of work; and iv) past application cases can suggest useful types of work substitutions for carrying out the adaptation. The user is asked to choose among the suggested types of work substitutions to be implemented. This adaptation mechanism has proved to produce acceptable solutions.

9.3.6- IMPLEMENTATION OF THE SYSTEM

Xtimela-CBR was implemented using Kappa-PC shell. Kappa-PC enabled the author to explore the combination of different reasoning approaches in a very complex domain, and therefore to learn about their strengths and weaknesses. The chosen implementation tool provided many of the means needed for developing Xtimela-CBR. However, Kappa-PC presented more programming limitations than other more powerful tools. The justification for using Kappa-PC was to privilege the modelling aspects of KBS models within the limited resources available for this research. These modelling aspects are independent of the programming details of any programming tool.

9.3.7- VERIFICATION AND VALIDATION OF THE SYSTEM

The V&V of the Xtimela-CBR was carried out using different methods. Each method focused on different aspects of the system's validity and utility. The V&V has shown that these methods tend to complement each other. The integration of the V&V within the system implementation and refinement was a very useful procedure, because it enabled the author to keep track of the system performance at every stage of the CCA method. The validation of the initial seed cases, treating them as new applications and validating each case added to the case library, has contributed to the successful predictive validation of the system. Predictive validation was regarded as a structured extension of the system's knowledge-base. Because of the large number of test cases, predictive validation was easy to carry out. The field and face validations have highlighted the limitations of the system's user interface. Filed validation was useful for keeping track of the Client's specific needs and ultimately for the Client's acceptance of the research project. The Client was in a position to see what the project could deliver and what it could expect from the potentialities highlighted during the validation of the system. The face validation was useful for capturing the views of independent experts on the system performance and on the model which supports the system. They provided useful contributions to the research.

9.4- SUGGESTIONS FOR FUTURE WORK

There are a number of possible extensions to the system developed in this research project. They are summarised below.

The results achieved in the system's validation, the interest shown by the Client and the encouraging response given by independent experts, indicate that Xtimela-CBR has the potential of being further improved and extended up to the stage of *embedded in use system* of the CCA method. In order to make such an extension, a number of improvements are required. They include:

- To proceed with case acquisition. It will be necessary to widen the case library with application cases covering new domain problems, such as: i) applications for common parts grants; and ii) applications for (HMO) houses in multiple occupation.
- To extend the case library organisation, in order to improve the performance of the case-base reasoner as the number of application cases grows. This can be achieved by adding further nodes to the case library organisation.
- To enhance the man-machine interface in general and in particular for: i) allowing the user to select the evidence of failures from a menu; ii) allowing the user to change both input and output information during the consultation; and iii) printing reports.
- To develop adequate documentation for supporting the Client in using and maintaining the system.
- To integrate the system within the Client's current information technology environment.
- To extend the system to cover other problem domains of the HRGS regarding type of applications, applicants, households and properties.
- To integrate the system with other presentation software packages which can enhance the communication between the user and the system.

The enquiry eligibility task can be extended as an independent help desk system to provide assistance over the telephone or directly to potential applicants for the HRGS. This help desk system should query, record and process the preliminary information provided by the potential applicant and, based on past enquiries, the system should suggest to the applicant a set of options for dealing with his/her needs, such as: i) the most appropriate type of grant for which to apply; ii) the likelihood chances of having a grant approved; iii) the predicted amount of a grant, if any; and iv) planning issues regarding the area in which the house is located.

The task of calculating the cost of schedules of work can be accomplished by using CBR. CBR can provide accurate estimates, by retrieving the costs of schedules of work from past application cases which have been successfully implemented, stored in the case library. This extension can be accomplished by extending the system's case-base reasoner component. Thus, the case-base reasoner should derive the cost of schedules of work by:

1. Taking the specification of each schedule of work as input and searching in the case library for a set of potential matching application cases containing that particular schedule of work which has been successfully accomplished in the past.

2. Searching and matching an application case which has the best price for a similar schedule of work to the current application case.
3. Retrieving the cost from the selected application case.
4. Repeating steps 1 to 3 for the set of schedules of work for the current application case.
5. Adjusting the retrieved costs to reflect current conditions.
6. Calculating the budget including fees and tax.
7. Reporting the budget to the user.

In terms of control, the task of calculating the cost of works should be accomplished after the schedules of work are known and the contractor has been selected.

To ensure that the works are carried out by competent and trustworthy contractors, the system can be extended to suggest a list of contractors who have good records. This additional task can be accomplished using CBR. This extension can be achieved by implementing the following steps:

1. Each application case should include three additional features: i) two to record the name and address of the contractor selected to carry out the works; ii) the third to record the performance achieved by the contractor in completing the works.
2. To create two additional partitions in the case library: i) one for application cases where works were carried out by bad contractors; and ii) another for application cases where works were carried out by competent and trustworthy contractors.
3. To extend the case-base reasoner such that it can derive a list of potential contractors.

The case-base reasoner accomplishes this extension by:

1. Taking the list of schedule of works and the area of the current property as input and searching the partition of application cases where the works were carried out by a competent and trustworthy contractor.
2. Matching a set of application cases with similar works and location area.
3. Retrieving the names and addresses of the contractors.
4. Reporting the list of contractors which are able to carry out the works to user.

In terms of control, the task of suggesting the list of contractors should be accomplished after the schedules of work are known.

From the development of Xtimela-CBR it seems that CBR can be useful for assessing applications for discretionary grants (renovation and disabled facilities). Application case prototypes can be used where real-world application cases are not available. The size and organisation of the application cases needs to be further investigated as the case library grows. One option is the organisation into case and sub-case hierarchies as suggested by Pearce, Goel, Kolodner, Zimring, Sentosa and Billington (Pearce M. et al., 1992). These authors suggested that this structure is suitable when the size of the cases is large.

Finally, it seems that CBR is a promising technique for developing KBSs in the construction industry. There are a number of application areas in the construction industry where CBR might be useful. They include:

Construction planning

Construction planning is a crucial and challenging management task. A good plan is essential to project success. Planning in construction is a knowledge-intensive task where construction planners use their past experience to produce plans for new construction projects (Zozaya-Gorostiza C., et al., 1989).

Several CBR applications in planning have been developed, or are under development (Kolodner J., 1993). These applications suggest that CBR can help to derive solutions to complex planning problems in the construction industry. Hammond (Hammond K., 1989) pointed out that "case-based planning is the idea of planning as remembering". A CBR system in construction planning can make use of its own past experience in developing new plans. It should rely on its memory of past construction plans. According to Hammond (Hammond K., 1989) case-based planning differs from rule-based planning in that it rests on the notion that new plans should be based on the planner's knowledge of what has succeed and failed in the past.

Selection of construction methods for building projects

Construction experts take decisions about construction methods during the construction stage of a building project. Selecting construction methods and materials involves two types of decisions. Firstly, the construction expert has to identify and choose among possible packages of labour and equipment available to build a building element. Alternate methods affect the resource requirements and consequently the construction cost. After the type of crew or technology package has been chosen, the second type of decision made by the construction expert is to select the number of machines or crews assigned to the construction activities.

In selecting the type and the number of technology packages or crews, the construction expert uses knowledge about crews or equipments usually chosen to perform an activity (Zozaya-Gorostiza C., et al., 1989). CBR suggests a model for collecting and representing technology packages which have been used in previous situations and to make them available for solving new problems. A CBR system to support construction experts in selecting construction methods should rely on its case library of past construction methods represented as cases. Solutions to new problems can be derived by: i) retrieving the most relevant previous technology package or crew; or ii) retrieving the most relevant previous technology package or crew and adapting it to fit the new situation.

Project appraisal

Generally the appraisal of a given construction project is a knowledge-intensive task within the project cycle. It provides a comprehensive evaluation of all aspects of the project and lays down the basis for implementing and evaluating it after completion. Appraising a construction project can be a lengthy process. It involves accessing and reviewing a variety of information sources, evaluating project alternatives, projecting a number of constantly changing economic parameters and significant judgements.

CBR combined with other techniques suggests a model for: i) developing a KBS to support human experts in appraising construction projects by taking advantage of the body of knowledge gained from past work; and ii) a way for dealing with the uncertain world of the construction projects. According to Kolodner (Kolodner J., 1993) CBR gives a reasoner a means of evaluating solutions when no algorithmic method is available for evaluation. A memory of past projects represented as cases can provide a basis for: i) evaluating project alternatives; and ii) suggesting project improvements. Thus, a KBS to support project appraisal may be able to accomplish two main functions: i) to act as a decision aiding system; and ii) to act as a shared external memory of past construction projects that allows those who use the system to share their experiences.

Construction management tutoring

CBR combining with multimedia technology suggests a way for teaching construction students.

A library of past construction projects that have the status of precedents can be a valuable source of teaching material. A project precedent can be an example of a particular type of construction project. Construction projects can be represented as precedent cases linked to appropriate illustrations. These precedent cases will be available to students who will be learning construction management.

Within these areas a number of CBR issues can be explored such as: i) CBR for deriving and evaluating new solutions; and ii) CBR's learning and tutoring capabilities.

In this section a number of suggestions have been presented which may be useful for future research in the construction industry.

APPENDIX 1

GLOSSARY OF TERMS USED IN THIS THESIS

Application or System: Reference to an information processing system which is built to perform a task or a set of tasks.

Case: A case is a contextualised piece of knowledge representing an experience that teaches a lesson fundamental to achieving the goals of the reasoner (Kolodner J., 1993).

Class: Abstract description of one or more similar objects.

Client: The Environmental And Consumer Services Department of the Salford City Council.

Control of processing: Means the selection of the operations to be performed at any given stage of the problem solving (Goel A., 1989).

Frame: A knowledge representation formalism consisted of one or more features represented as slots.

Knowledge-base: It is the repository of knowledge in a computer system.

Knowledge-level: A knowledge modelling level for describing intelligent system as having goals, actions and bodies of knowledge (Newell A., 1982).

Means test: It is the procedure carried out to test the financial resources of the relevant persons to a given application for the HRGS.

Net Present Value: It represents the assessment of the costs and benefits of a given action in present value terms.

Object-oriented programming: A wide range of programming techniques. Its building blocks are objects, inheritance, methods and functions. Objects can communicate each other through methods.

Object: It is a knowledge representation formalism consisted of one or more features represented as slots and one or more methods.

Problem solving method: It is consisted of a set of sub-tasks that can be used to transform the initial state of a task to the goal state (Chandrasekaran et al., 1992).

Problem space or Search space: Is defined by a set of states reachable from the initial state by applying the sub tasks for the problem solving method (Chandrasekaran B. et al., 1992).

Production rules: An item of knowledge which takes the form "*if* condition *then* a action is appropriate".

Search control knowledge: It is the knowledge that guides the search for a solution through the problem space, i.e., is the knowledge need for sequencing the sub tasks (Chandrasekaran B. et al., 1993).

APPENDIX 2

ABBREVIATIONS USED IN THE THESIS

AI- Artificial Intelligence
BRE- British Research Establishment
BS- British Standards
CBR- Case-Based Reasoning
CCA- Client Centred Approach
CIRIA- Construction Industry Research And Information Association
DFG- Disabled facilities grant
DoE-Department of Environment
DTI- Department Of Trade And Industry
EHCS91-English House Condition Survey 1991
HMO-Houses in multiple occupation
HRGS- House Renovation Grant System
IT- Information Technology
KBS- Knowledge Based System
NBS- National Building Specification
NPV- Net Present Value
PC- Personal Computer
PSA- Property Services Agency

APPENDIX 3

HOUSE RENOVATION GRANTS	UNIVERSITY OF SALFORD
QUESTIONNAIRE TO LOCAL HOUSING AUTHORITIES	DEPARTMENT OF SURVEYING

This questionnaire is organised in two sections. The first section focuses on the assessment process of applications for House Renovation Grants and the implementation of the renovation grant system. The second section requests your views on various issues, based on your experiences and perceptions of the House Renovation Grants System.

SECTION 1 : ASSESSMENT OF ENQUIRIES FOR HOUSE RENOVATION GRANTS

This section is concerned with the processing the enquiries for house renovation grants.

1 What does the service provided to grant applicants include ?

	please tick
preliminary assessment of grant eligibility	
property survey	
preparation of schedule of eligible works	
test of financial resources	
assistance with completing the enquiry	
list of contractors	

2 What is the average workload regarding the implementation of the renovation grant system in terms of the number of applications per month?

	0 to 50	51 to 100	101 to 200	201 to 300	351 to 500	more 501
please tick						

3 How much of your workload(applications) is by type of property ?

approx. percentage by numb. of applications

	0-20	21-40	41-60	61-80	81-100%
Terraced					
Mid-Terraced Dwelling					
End Terraced Dwelling					
Semi-Detached Dwelling					
Detached Dwelling					
Flat(s)					
Cottage					
HMO					

4 How much of your workload(applications) is by condition of the property ?

approx. percentage by numb. of applications

	0-20	21-40	41-60	61-80	81-100%
unfit for human habitation					
fit for human habitation					

5 How much of your workload(applications) is by property age ?

	approx. percentage by numb. of applications				
	0-20	21-40	41-60	61-80	81-100%
pre-1919					
1920-1929					
1930-1944					
1945-1964					
1965-1984					
post-1985					

6 How much of your workload(number of applications) is according area based schemes?

	approx. percentage by numb. of applications				
	0-20	21-40	41-60	61-80	81-100%
Inside a group repair scheme					
Inside an area renewal area					
Inside a clearance area					
Outside any area based scheme					

7 How much of your workload(number of applications) is by type of applicant ?

	approx. percentage by numb. of applications				
	0-20	21-40	41-60	61-80	81-100%
Owner Occupier					
Landlord					
Local Authority Tenant					
Private Tenant Requi.to Carry Out Repairs					
Private Tenant Not Req.to Carry Out Rep.					
Housing Association Tenant					
Prospective Purchaser					
Participating Landlord					
Long Leaseholder					
other					

8 What are the types of grant most applied?

	approx. percentage by numb. of applications				
	0-20	21-40	41-60	61-80	81-100%
Renovation Grant					
Minor Works Assistance					
Common Parts Grant					
Disabled Facilities Grant					
HMO Grant					

9 What is the type of grant approved?

	approx. percentage by numb. of applications				
	0-20	21-40	41-60	61-80	81-100%
Renovation Grant					
Minor Works Assistance					
Common Parts Grant					
Disabled Facilities Grant					
HMO Grant					

SECTION 2 : IMPLEMENTATION OF THE HOUSE RENOVATION GRANTS SYSTEM

Here we would like your views on the implementation of the House Renovation Grants system.

10 What are the main steps involved in the processing of an application for the HRGS grants ?

	please tick
preliminary assessment of grant eligibility	
property survey	
preparation of schedule of eligible works	
test of financial resources	
calculation of the amount of grant	
other(s) ______	
______	
______	

11 Do you use experience gained from past work for the processing of new applications?

	a lot	sometimes	occasionaly	rarely	not at all
please tick					

12 Do you find any similarities or analogies among the applications ?

	strong		some		none
please tick					

13 Do you use any particular guidance or framework for the assessment of applications?

	yes	no
please tick		

If yes to the question 13 , please can you provide a copy in your reply.

14 Do you use the computer to help the processing of applications ?

	yes	no
please tick		

15 If yes to the question 14 , please specify for what tasks do you use the computer and the software .

task	software used

15 How happy are you with the existing software ?

	very happy	happy	neutral		not at all
please tick					

16 How do you think decisions taken on past applications can be useful to guide in deciding

on new applications?

	very useful	useful	some		not at all
please tick					

17 Details of person completing questionnaire:

Name: ______________________________

Position: ______________________________

Local Authority: ______________________________

____

address: ______________________________

____

____

Thankyou for completing this questionnaire , please return it direct to Salford University in the S.A.E. provided . Alldetails will be dealt with in strict confidence .

Francisco Ribeiro
University of Salford
Department of Surveying
Bridgewater Building
Salford M5 4WT

APPENDIX 4

KNOWLEDGE-LEVEL MODELLING METHODS

Several different methods for knowledge modelling have been developed by researchers working in Artificial Intelligence. The majority of these methods are designed to describe KBS in terms of the knowledge they have, rather than by the details of their implementation. Also, they share the view that the knowledge must be modelled with respect to a goal or task.

A detailed description of each modelling method listed in table 4.1 is presented below. This description is made based on the main features of each approach.

1. Heuristic classification (Clancey W., 1985, 1993)

Clancey in his early work, found that classification problem solving is a common phenomenon and a major information processing task that occurs within a number of KBSs for diagnosis. He identified what he called *heuristic classification* a problem solving strategy for the diagnosis task which is mapping data into categories. Clancey found that MYCIN inference engine is an example of heuristic classification.

According to Clancey (Clancey W., 1985, 1993), *heuristic classification,* can be decomposed into three sub-tasks or inference types: *data abstraction, heuristic match,* and *solution refinement* as shown in the figure.

Clancey proposed that diagnostic strategies in the heuristic classification are a collection of micro-tasks organised in a certain order, and that these micro-tasks can be achieved by appropriate met-rules in a rule-based system. Finally he developed a language called heracles that directly supports the specification of knowledge needed for the micro-tasks, invoking them and combining to create the heuristic classification behaviour for diagnosis. Later, Clancey developed a diagnostic model for NEOMYCIN that comprises diagnostic strategy and a structure of the task.

The main claim of Clancey's diagnostic strategy is that diagnosis can be described abstractly as a process in which the problem solver poses tasks for himself in order to have some structuring effect on working memory. Metarules for performing a task bring appropriate sources of knowledge to mind. Thus, it is important that the procedure is structured so that the tasks make sense in the same way that people do. These procedures are abstract because they are separated from the domain knowledge and are explicit because they are well structured.

2. Inference nets (Brandon P. et al., 1988; Giarratano J. and Riley G., 1994)

Formally, an inference net can be defined as directed cyclic graphs where nodes are assertions or hypotheses or evidences or states and, arcs establish relationships or actions or causal connections between nodes.

According to Giarratano and Riley (Giarratano J. and Riley G., 1994) an inference net knowledge provides a tool to knowledge modelling and is useful for two purposes: i) it helps to organise knowledge about a subject by classifying objects or concepts and showing their relationships to other objects or concepts; and ii) it helps to guide the search for a proof of a hypothesis. Inference nets have a static knowledge structure. That is, the nodes and connections between them are fixed in order to retain the relationships between nodes in the knowledge structure. To each node can be associated certainty factors or probabilities.

3. KADS methodology (Wielinga B. et al., 1993)

A fundamental premise of the KADS approach is that development of KBSs should be driven by the demands and structure of the task, rather than by the demands and structure of any particular implementation model. According to Wielinga, Schreiber and Breuker (Wielinga et al., 1993) the principles that underlie the KADS approach are:

1. The introduction of multiple models as a mean to cope with the complexity of the knowledge engineering process namely: i) the *organisational model*; ii) the *application model*; iii) the *task model*; iv) the *model of co-operation*; v) the *model of expertise*; vi) the *conceptual model*; and vii) the *design model.*
2. The use of knowledge-level descriptions as an intermediate model between expertise data and system design.
3. The re-usability of generic model components as templates supporting top-down knowledge acquisition.
4. The process of differentiating simple models into more complex ones.

5. The importance of structure-preserving transformation of models of expertise into design and implementation.

The KADS methodology view the task model, the model of co-operation and the model of expertise as being knowledge-level models that specify the task, sub-tasks and the desired problem solving behaviour for the KBS. Two basic ideas support the four-layer framework for modelling:

- Firstly, it assumes that it is possible and useful to distinguish between several generic types of knowledge according to different roles that knowledge can play in the reasoning process.
- Secondly, it assumes that these types of knowledge can be organised into several layers, which have only limited interaction.

The four-layer framework distinguishes four categories in which the knowledge can be analysed and modelled: i) *static knowledge* describing a declarative theory of the application domain (domain knowledge); ii) *knowledge of different types of inferences* that can be made in this theory (control knowledge); iii) *knowledge representing elementary tasks* (control knowledge); and iv) *strategic knowledge* (control knowledge).

4. Role-Limiting Methods (McDermott J., 1988, 1993)

McDermott (McDermott J., 1988, 1993) investigated the roles of knowledge in various problem solving methods and tasks and as result they developed the role-limiting methods approach for solving several general tasks such as diagnosis and design. According to McDermott (McDermott J., 1993), role-limiting methods are "problem solving methods that strongly guide knowledge collection and encoding". They have a broad scope of applicability and provide help in specifying what knowledge needs to be collected to perform a particular task and how that knowledge can be properly encoded. The approach specifies the roles that various types of knowledge play in the operation of each problem solving method. The problem solving method is the central key in understanding and building an application. Each problem solving method contains certain roles that need to be filled by domain knowledge, i.e., they describe what domain knowledge is expected from the available knowledge sources.

McDermott and his colleagues developed an automatic knowledge acquisition tool for collecting and encoding the knowledge required by different role-limiting methods. According to the authors, role-limiting methods provide several advantages:

- the knowledge acquisition tool needs a clear notion of the function of the knowledge required by the system it is building so that it knows what information to acquire;
- a functionally represented knowledge base can be examined to judge the expert system's adequacy to perform the task for which it is intended;
- understanding the roles knowledge plays allows the knowledge acquisition tool to generate a problem solving system that knows how to apply the gathered knowledge when appropriate;
- in order for a KBS to describe to a user how it makes its decisions, an explanation facility must have an understanding of the function of the knowledge the system uses; and
- understanding of the knowledge roles during problem solving is crucial in mapping a domain expert's problem description onto a problem strategy.

Examples of role-limiting methods include: i) *Cover-and-differentiate-* a method suitable for certain types of diagnostic tasks. This method assumes that the initial state is a set of one or more symptoms. ii) *Propose-and-revise-* a method suitable for certain types of constructive tasks. The method assumes that the initial state is a set of specifications. iii) *Qualitative reasoning-* a method suitable for certain types of diagnostic tasks. iv) *Acquire-and-present-* a method suitable to report. v) *Extrapolate-from-a similar-case.*

5. Componential Framework (Steels L., 1990)

Steels (Steels L.,1990) proposed the componential framework (CF) for modelling knowledge at knowledge-level and knowledge-use level. The componential framework underlies the notion of: i) tasks; ii) task structures (sub-tasks and problem solving methods); and iii) models used by the human experts. The Steel's CF describes the task decomposition at different levels allowing one to make a mapping from conceptual features, pragmatic constraints of a task, and available knowledge to components such as: *problem solving methods, domain models, and task structures*. It recognises that the sub-tasks of a task depend on the problem solving method used for the task. The system's description according to the Steel's CF, move in a top-down systematic fashion and includes the following components:

Task analysis: Firstly, there is a detailed analysis of the task (real-world task). The task can be decomposed into sub-tasks with input-output relations between them. Each task and sub-task is analysed in terms of the problem that needs to be solved (based on the description properties of the input, the expected output, and the nature of the operations taking place to map input to output, the generic task, the pragmatic constraints in the task that result from the environment in which the system will operate or from the limitations that humans have, the case model for the task and the available knowledge).

Problem solving methods: The next step, is to investigate the set of possible problem solving methods and associated domain models that either decompose the task into sub-tasks or directly solve the task. According to the Steel's framework problem solving methods are responsible for applying domain knowledge to a task.

Task structure: The final step, given the task decomposition, instantiate the task implied by the method. Reiterate for each of the sub-tasks starting from the first task until tasks have been reached that are directly solved by the application of domain knowledge.

6. Task Structure Analysis(Chandrasekaran B. et al., 1992)

By the early 1980s Chandrasekaran and his colleagues started formulating the notion of a generic task. Generic tasks identify a type of task, a problem solving method for doing that task, and the kinds of knowledge needed to use the method.

Each generic task is characterised by information about the following (Brown D. Chandrasekaran B., 1989):

1. *The type of problem* (task specification in form of generic types of input and output information) (the type of input and the type of output). What is the function of the generic task? What is the generic task good for?
2. *The representation of knowledge*. How should knowledge be organised and structured to accomplish the function of the generic task? In particular, what are the types of concepts that are involved in the generic task? what concepts are the input and output about? How is the knowledge organised in terms of concepts?
3. *The inference strategy* (process, problem solving, control regime). what inference strategy can be applied to the knowledge to accomplish the function of the generic task? How does the inference strategy operate on concepts.

What Chandrasekaran's generic task really meant was: an elementary generic combination of a problem, representation, and inference strategy. If a problem or a sub-problem matches the function of a generic task, then the generic tasks for that problem provides a knowledge representation and an inference strategy that can be used to solve the problem or task at application level. Chandrasekaran and his colleagues have developed some of the generic tasks that can be used to diagnosis and design problems. Examples of these generic tasks are: i) *for diagnosis task*: hierarchical classification, hypothesis matching or assessment, knowledge-directed information passing, abductive assembly, concept-matching and data abstraction for diagnosis; and ii) *for design*: hierarchical design by plan selection and refinement, and state abstraction.

With experience gained with generic tasks and in response to the need for uniform framework, Chandrasekaran and his group developed the **task structure analysis**, what they considered a uniform task-level analysis for describing a knowledge based system at knowledge level.

The task structure analysis produces a task structure, i.e., a tree of tasks, alternative problem solving methods, and sub-tasks applied recursively until tasks are reached that are in some sense performed directly using available knowledge for a given problem. In general, a task can be accomplished using any one of several alternative problem solving methods. Thus the task structure should explicitly identify alternative problem solving methods for each task. A method can set up sub-tasks, which themselves can be accomplished by various alternative problem solving methods. In this view a method is viewed as a way of accomplishing a task. Problem solving methods and tasks are not completely independent. The task structure comprises the following components:

Tasks: Tasks are specified as transforming an initial problem state with certain features to a goal state with certain additional features. For example, in the case of the applications for renovation grants (problem), regarding the task of assessing the fitness of a house the initial state include the observed condition of the house regarding the fitness standard and the goal state includes information on the fitness of the house, and its grounds of unfitness if deemed unfit. The Chandrasekaran methodology makes distinction between a task and a task instance. A task instance is a particular problem. In contrast, a task specifies a family of task instances of a certain type. For example the fitness assessment of a property is an instance of the task of the condition assessment of buildings, which is itself is a subclass of the general diagnosis task.

Problem solving methods and sub-tasks: The problem solving methods in the task structure are regarded as ways of accomplishing tasks and may be of many types. Chandrasekaran (Chandrasekaran B., 1990) has classified the problem solving methods into two types: i) one type are those methods that can be viewed as a search in a problem space;

and ii) the other type are those methods (consisting of compiled algorithms) that directly produce a solution within any search in a space of alternative solutions. Those methods that are modelled as problem space search (first type of problem solving methods), are defined in the uniform framework as being a set of sub-tasks that can be used to transform the initial state of a task to the goal state. The method may contain the knowledge required to sequence the sub-tasks- called search control knowledge.

Required Knowledge: The knowledge required by the problem solving methods in the task structure functionally comes in four different types: i) *knowledge to accomplish each task* that maps the input the task to its output (goal); ii) *knowledge to indicate when a sub- task is needed*; iii) *knowledge to sequence the sub-tasks* when they are required by the method; iv) *knowledge needed to select a method* when there are alternative problem solving methods available to the task. The four types of knowledge in a task structure can be available in two forms: i) it can be directly available in the system for the task (that maps the input of the task to its output) or ii) if the knowledge is not direct available it can be derived from the existing knowledge or, can be acquired from external sources, or can be computed using domain models (causal models). In both cases a problem solving method must be used.

The task structure is meant to be an analytical tool that provides a vocabulary to use in describing how the system works and to map the knowledge acquisition and system design. Hence, the system being described might be based on rules, frames, cases or logic predicates, i.e., using the available technology. This is unimportant for the use and construction of the task structure. Chandrasekaran, Johnson and Smith (Chandrasekaran B. et al., 1992) proposed a task structure for the diagnosis and design tasks. Part of the task structure diagram for the design task is shown in the figure below. In the diagram circle represent tasks and rectangles represent alternative methods.

7. Problem solving methodology (Puerta A. et al., 1992)

Puerta, Tu and Musen (Puerta A. et al., 1992) developed the problem solving methodology, which uses tasks, problem solving methods, sub-tasks and mechanisms as mediating concepts for task analysis and modelling. The product of the task analysis and modelling is a hierarchy of concepts structured according to the following levels:

- at the top level there is a **task**: this is an activity in the real world;
- then there is the **problem solving method**, which is a 'procedure that implements an abstract model of problem solving and that is applicable to a class of tasks;

- below the method is the **sub-task**, identical to a task except in its occurrence in task decomposition's; and
- finally there is the **mechanism,** which is a method that does not decompose a task into sub-tasks but is used to accomplish the task (i.e. it is at the bottom of the tree).

8. Knowledge Level Primitive (KLP) (O'Hara K. and Shadbolt N., 1993)

The knowledge level primitive is a task-oriented approach where the knowledge primitive is the key mediating concept used in task analysis and modelling. The approach attempts both to conceptualise a given problem in the real world, and to constrain the problem solving methods appropriate for that problem. According to the KLP methodology a **task** is viewed as a real world conceptualisation of a problem and a **method** is a conceptualisation of a problem that renders its solution. The KLP methodology produces a configuration of knowledge primitives that can attach themselves firmly both to tasks and methods. This configuration account for three levels such as: the task level called by task space; the knowledge primitive level called by KLP space; and finally the method level also called the method space. The KLPs are indexed by the methods they legitimise thus they can be defined by the methods.

9. The Knowledge Level Computational Model (KLCM) (Smith J. and Johnson T., 1993)

The knowledge level computational model KLCM, describe a knowledge based system in terms of the *task or goal* it is to perform, the *actions* it can use to perform the task, and the *knowledge* it has to select actions and the task environment. The KLCM description is implementation-independent: it only specifies what the system should do. The only requirement for describing a system at KLCM is that it be viewed as having goals, actions, and a body of knowledge. From KLCM point of view, a system is an idealised agent interacting with an environment. The agent has goals, actions, and environment.

APPENDIX 5

TASK STRUCTURE FOR THE ASSESSMENT OF APPLICATIONS FOR RENOVATION GRANTS

5.1- NEW APPLICATION CASE GENERATION

Definition: New application case generation task is specified by a set of case representation features.

Initial state: The initial state is a set of data which includes: application code; applicant's name; property address; and type of grant sought by the applicant.

Goal state: The goal state is the one containing a specification of a new case for the current application, i.e., a case representation appropriate for the grant sought by the application.

Problem solving method: The new case generation can be seen as a selection problem where the initial data of the current application case is matched it with the specifications of the case representations. Therefore, knowledge needed for the search and matching can be formulated in terms of rules. Thus, the associative method was selected to accomplish this task.

Sub-tasks: The associative method decomposes the new case generation task into three sub-tasks: i) *to ask data*; ii) *to select an application case representation*; and iii) *to create a new application case.*

To ask data has the goal of acquiring the preliminary information required to evaluate the enquiry eligibility.
To select an application case format has the goal of selecting a case representation appropriate to the new application.
To create a new case application has the goal of creating a new case for the current application.

Sub-task proposal knowledge: Firstly, to apply to *select an application case representation* when known: the application code; applicant's name; property address; and type of grant sought. Secondly, to apply to *create a new case application* when the case representation for the current application is known.

Search control knowledge: The associative method accomplishes the new generation task by: Firstly, by matching the initial data of the new application with specifications of the stored case representation categories. Secondly, by creating an instance of the matched case representation category for the current application in the case library.

Figure A5.1 shows part of the task structure for the task application case generation.

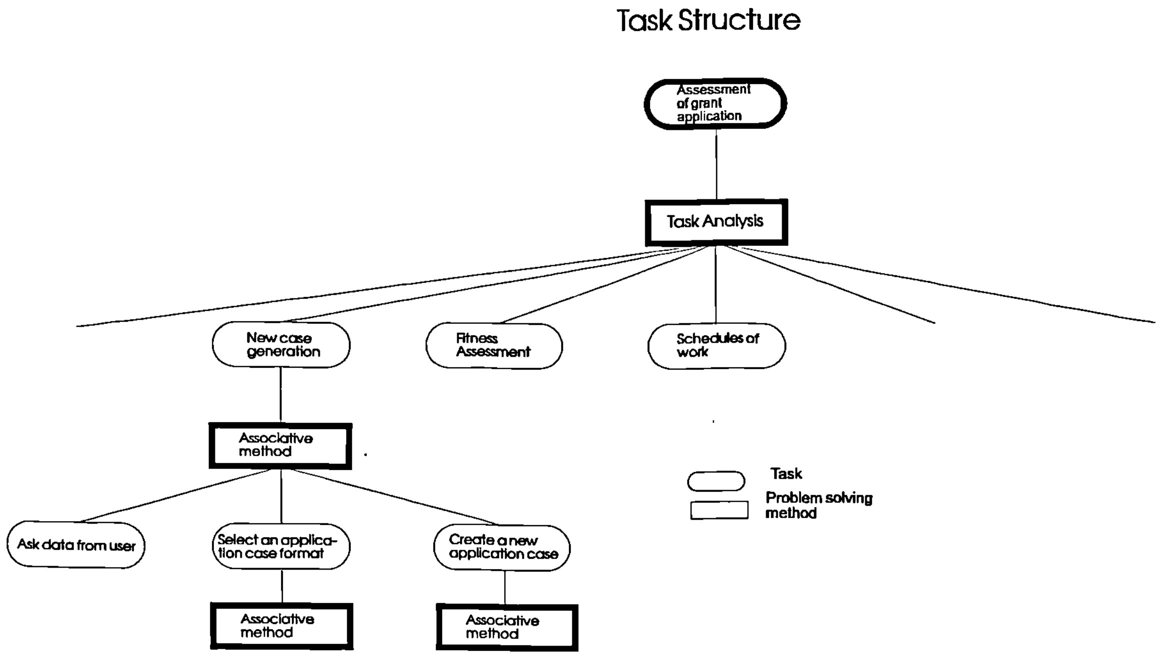


Figure A5.1: Part of the task structure for the task application case generation.

5.2- ENQUIRY ELIGIBILITY

Definition: A preliminary enquiry should be carried out before a formal application is made in order to minimise the number of 'abortive' applications. The enquiry eligibility task is specified by a set of conditions governing the enquiry eligibility for a grant application.

Initial state: The initial state includes a set of data about the enquiry, such as: age of property; interest of the applicant in the property; type of certificate of future occupation that will be provided; applicant type; purpose of works; state of works; type of use of the residence; grant past history on the property; and issues regarding Building Regulations approval and planning permission.

Goal state: The goal state includes: i) the eligibility conditions that have been satisfied; and ii) eligibility of the enquiry specified in terms of proceeding with an application or refusing a grant.

Problem solving method: The associative method was chosen for evaluating the enquiry eligibility.

Search control knowledge: The associative methods accomplish the task by: Firstly, asking data about the enquiry. Secondly, matching the enquiry data with the conditions required to make an enquiry to be eligible for the specified grant. Thirdly, evaluating the eligibility considering the number of conditions matched by the new the enquiry. If all conditions are satisfied then the enquiry is eligible to proceed.

Domain Model: One domain model was used for representing the knowledge about the grant eligibility conditions.

Figure A5.2 shows part of the task structure for the enquiry eligibility task.

5.3- COST OF WORKS

Definition: The cost of works is specified by a set of schedules of work and the selected contractor estimate.

Initial state: The initial state includes information about: i) the specification of schedule of works; and ii) a contractor estimate.

Goal state: The goal state is the one which includes the costs of the approved works.

Problem solving method: The cost of works is determined by calculating an algorithm. This algorithm performs the cost of works by: i) assigning an approved cost to each schedule of work; ii) computing the total cost of works.

Figure A5.2 shows part of the task structure for the cost of works task.

5.4- ECONOMIC ANALYSIS

Definition: The economic analysis is specified by: i) the net present values of alternative courses of actions available to deal with the unfit property; and ii) the limit of £20,000 for the total costs of works eligible for a mandatory renovation grant.

Initial state: The initial state includes information about: i) attributable costs and benefits of renovation option; ii) attributable costs and benefits of alternative options to the renovation; and iii) limit of total cost for renovation works.

Goal state: The goal state includes the comparative merits (cost and benefits in terms of net present values (NPVs) of the renovation works with respect to a set of alternative options.

Problem solving method: The associative method was chosen for the task.

Sub-tasks: The associative method accomplishes the economic appraisal by decomposing the economic analysis task into two sub-tasks: i) *to calculate NPVs of each course of action* (costs and benefits); ii) *to determine the merits of renovation option.*

Sub-task proposal knowledge: Apply the task *to calculate NPVs* when the attributable cost and benefits of each option are known. Apply the task *to determine economic merits of renovation option* when the NPVs of each alternative option are known.

Search control knowledge: The associative method perform the task by: i) calculating the NPVs of each options following an economic formula: ii) comparing the NPV of renovation with the NPVs of other options; and iii) matching with the merits stored in the system.

The description of each sub-task mentioned above follows in the next sections.

5.4.1- Calculate NPVs Of Available Course Of Actions

Initial state: The initial state includes the information about: i) attributable costs and benefits of renovation option; and ii) attributable costs and benefits of the alternative options.

Goal state: The goal state includes the assessment of cost and benefits in terms of NPVs of the available courses of action: i) renovation (making the property fit); ii) demolition and clearance; iii) demolition and rebuild; and iv) maintaining the condition of the property as it is.

Problem solving method: An algorithm was chosen for the task.

5.4.2- Assess The Merits Of The Renovation Option

Initial state: The initial state includes the information about: i) the NPVs of each courses of action available; and ii) limit for the total cost of the renovation works.

Goal state: The goal state includes the assessment of the comparative merits of the renovation option.

Problem solving method: The associative method was chosen for the task.

Search control knowledge: The associative method performs the task by: i) comparing the NPV of renovation action with the NPVs of alternative actions and the cost limit; and ii) matching the comparison to the merits stored in the system.

Figure A5.2 shows part of the task structure for the economic analysis tasks.

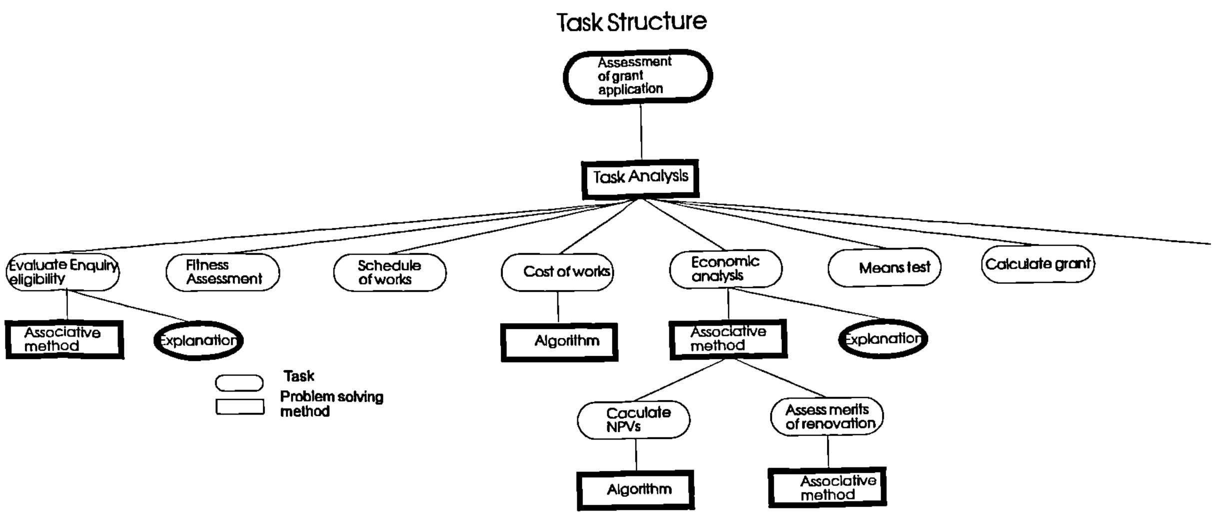


Figure A5.2: Part of the task structure for the enquiry eligibility, cost of works and economic analysis tasks.

5.5- CALCULATION OF GRANT

Definition: The calculation of grant is specified by the cost of works and the reduction in the amount of grant.

Initial state: The initial state includes information about: i) the total approved costs; ii) the reduction of grant; and iii) the approved contractor estimate.

Goal state: The goal state includes the amount of grant.

Problem solving method: An algorithm was chosen for the task.

5.6- DEGREE OF ELIGIBILITY

Definition: The degree of eligibility is defined by a criteria.

Initial state: The initial state includes information about: i) the housing needs of the occupants; ii) the condition of the property; iii) the provision of certificate of future occupation; iv) the past assistance on the property; v) the other special needs of the occupants; vi) the amount of grant.

Goal state: The goal state is the one which includes the degree of eligibility for the current application.

Problem solving method: The associative method was selected for the task.

Task proposal knowledge: Apply the task when the data about the initial state is known.

Search control knowledge for sequencing the operators: The associative method accomplishes the reduction of grant by matching the initial state data with the degrees of eligibility stored in the system.

Figure A5.3 shows part of the task structure for calculating the amount of grant and the degree of eligibility.

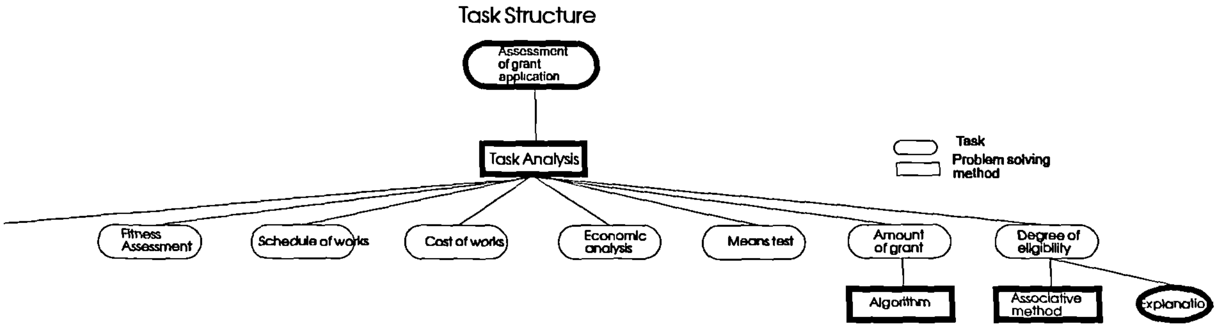


Figure A5.3: Part of the task structure for calculating the amount of grant and the degree of eligibility

APPENDIX 6

KNOWLEDGE ACQUISITION METHODS

1.Problem solving method-based (McDermott J., 1988; Birmingham W. and Kinkler G., 1993)

Examples of problem solving method-based approaches for knowledge acquisition are: i) the role-limiting method from McDermott (McDermott J., 1988); and ii) the problem solving model from Birmingham W. and Kinkler G. (Birmingham W. and Kinkler G., 1993). These methods use information from the problem solving method to drive the knowledge acquisition. The problem solving methods determine the knowledge roles that must be filled with domain knowledge. The knowledge roles determine the structure of the knowledge-base, and establish a vocabulary that can be used to guide the knowledge acquisition. Based on these methods several tools for knowledge acquisition have been developed, such as: Salt (McDermott J., 1993); and Knack (Klinker G. et al., 1987).

According to Yost (Yost G., 1993) the method-based knowledge acquisition approaches comprise some basic assumptions:

- a problem type; the class of applications that the method can be used to implement; and
- a problem solving method that can perform tasks of this type. The method has knowledge roles that must be filled with domain knowledge.

2.Task-based method (Chandrasekaran B., 1986a, 1990)

The task-based method, uses information about a task-specific problem solving method, given by the generic task, to drive the knowledge acquisition. The task-specific problem solving method identifies the types of knowledge required by the method and a vocabulary, which are used to guide knowledge acquisition.

3.Problem spaces (PSCM) (Yost G., 1993)

Problem spaces comprise states, operators, and the knowledge that relates to them. In this approach, a task can be performed by a single problem space or by a co-operating set of problem spaces. Tasks are performed in problem spaces. Knowledge roles are apparently in the process of performing tasks in the problem spaces. Each kind of problem space has a number of well-defined knowledge roles. The possible relationships among problem spaces co-operating correspond to these knowledge roles, since the only service one problem space can perform for another is to supply knowledge that is missing from one of its roles. These knowledge roles are used to guide knowledge acquisition.

4.KADS (Schreiber G. et al., 1993; Wielinga B. et al., 1993)

The principle of multiple models and the existence of four knowledge categories underlies the knowledge acquisition using KADS approach. The four knowledge categories (domain, inference, task and strategic) can be viewed as four levels with meta-like relations in the sense that each successive level interprets the description at the lower level. This four-layer framework is used as a basis for structured knowledge acquisition.

5.Interviewing (McGraw K. and Harbison-Briggs K., 1989; Shaw M. and Woodward J., 1990)

Interviewing assumes that knowledge expressed by expert is based on the question prompts and is used by the expert during task completion (Shaw M. and Woodward J., 1990). This method uses three elicitation procedures:

1. Unstructured interviews are appropriate when one wants to explore an issue or goal. It facilitates the description of the domain.
2. Structured interviews force the organisation of the interview.
3. Focused interviews are semi-structured interviews with some focused issues.

Interviews protocols, usually require a retrospective approach by the expert.

6.Protocol Analysis (Shaw M. and Woodward J., 1990)

This method includes both observations of expert behaviour and expert verbalisations. An example of this method is the *Specification of Expertise* (Shaw M. and Woodward J., 1990).

7.KA- Client Centred Approach (Watson I. et al., 1992b)

Watson, Basden and Brandon (Watson I. et al., 1992b) in their Client Centred Approach (CCA) for expert system development introduced other model for knowledge acquisition which involves obtaining a structured and understandable expression of the expertise that can be encapsulated in the expert system. Their method sates "it is insufficient merely to elicit problem solving heuristics because these are subjective and context sensitive, being composed of underlying understanding (U) and Context Dependent Problem Solving (CPS)". The authors stress that the knowledge acquisition process should separate the U from CPS element.

8.Learning-based techniques (Shaw M. and Woodward J., 1990)

There are several learning-based techniques. Examples of these techniques are:
Analogy: apply knowledge from old situations in similar new situations.
Apprenticeship learning: learn by watching experts solve problems.
Decision tree induction/analysis: generate, analyse decision trees.
Example selection: select an appropriate set of examples fro various learning techniques.
Similarity-based learning: learn similarities from set of positive examples and differences from sets of negative examples.

9.Text knowledge acquisition and analysis (Regoczei S. and Planting E., 1988; Moulin B. and Rousseau D., 1992; Anick P., 1993)

Regoczei and Plantinga (Regoczei S. and Planting E., 1988) argued that knowledge acquisition from texts is essential natural-language understanding, since "words create knowledge by causing a cognition an agent to form new mental models or alter existing mental models". Later Moulin and Rosseau (Moulin B. and Rosseau D., 1992), proposed an approach to knowledge acquisition and analysis from regulatory texts. In their view a regulatory text is like a natural-language knowledge base that describes a practical domain to which the law applies that contain three types of propositions: i) definitions; normative propositions; and meta-textual statements. Their approach is based on the assumption that prescriptive texts such NBC are written systematically by applying a domain-independent meta-language based on micro structure components. The micro structure is considered as the logical content, ,such headings, titles, chapter organisation, sections, and articles. This method, uses text grammars to analyse the macro structure and the micro structure components and their relationships. The grammar uses syntactic markers such modal operators (verbs), connectors (conjunctions), internal references, and punctuation to recognise the relevant objects of the standard world, the rules that apply to them, and the context of their application.

10.Case Acquisition Methods (Reisbeck C. and Schank R., 1989; Skousen R., 1989; Beck H., 1991; Ashley K. and Aleven V., 1992; Kitano et al., 1993; Kriegsman M. and Barletta R., 1993)

Riesbeck's Direct Memory Access (DMAP), is referred as a case-based parser, which views parsing as a problem primarily of memory search, given cues from the input text (Reisbeck C. and Schank R., 1989). It goes directly from words to domain-specific memory nodes. Hence in DMAP, the use of cases is primarily for parsing.

Skousen (Skousen R., 1989), proposed the analogical model where language understanding is presumed to be based on analogy on opposition from supposing a purely rule-based approach to understanding.

Beck (Beck H., 1991), proposed a case-based language understanding from similar cases in the context of concept acquisition and category theory and design as information retrieval method. According to Beck, the case-based language understanding is a "language usage" theory balance similarity-based reasoning with rule-based reasoning. The role of case-based reasoning in understanding unusual language is emphasised.

The dialectical examples method introduced by Ashley and Aleven (Ashley K. and Aleven V., 1992), generate argument contexts, collection of cases that instantiate dialectical examples from database of cases according user's general specifications. The approach was designed for tasks, such tutoring and legal reasoning. Kitano, Shimazu and Shibata (Kitano et al., 1993), introduced the Case-Method, a methodology to build large scale case-based systems, that underlies an approach to case acquisition which involves the following phases:

1. *Case collection-* requires collecting seed cases. The seed cases provide initial concept regarding the application domain landscape. The seed cases provide a crude case format and data structure As a start up phase, the cases are generally collected in unstructured and uninformed style, such as full-text and other domain-specific forms. This phase involves the following steps: i) collection of cases which are consistent with pre-defined case report format, and ii) filtering of cases so that only cases with minimum acceptable quality will be sent to the next phase. Cases are reported in structured style, using pre-defined case report form and full-text with specified writing style. Products of this phase are: i) a set of case report forms, and ii) a set of case reports in full-text.
2. *Attribute-value extraction-* the of the attribute-value extraction phase is to extract all possible elements in case representation and indexing. In the initial cycle, this phase consists of three steps: i) keyword listing; ii) attribute identification and; value grouping.

Each attribute and value are examined, to determine whether or not it is independent from other attribute values. Ideally, a set of attributes is expected to be a linearly independent set. However, in reality, this is not always possible, Thus, some dependency would be allowed. Products of this phase are: i) a list of attributes; ii) a list of possible values for each attribute; iii) a thesaurus of keywords to be the value of each attribute; and iv) a set of normalised units for problem description and evaluation.

3. *Hierarchy formation-* the hierarchy formation phase defines relationships among keywords and attributes. For each attribute identified in the previous phase, a set of keywords has already been grouped. In this phase, relationships between keywords will be defined. a product of this phase is a set of concept hierarchies created fro each attribute. The hierarchies created for each attribute. The hierarchies are assigned with similarities between values.
4. *Database definition and data entry-* case database definition will be created using the set of hierarchies defined in the previous phase. The products of this phase are a new case report format to be used for reporting cases in the next cycle.

Kriegsman M. and Barletta R. (Kriegsman M. and Barletta R., 1993), have considered four basic steps in case acquisition, such as: i) firstly, consisting of collecting data; ii) secondly, consisting of extracting features from data; iii) thirdly, consisting of indexing the data; and iv) finally, testing and refining the indexing scheme.

- From the above description, it seems that:
- there are methods suited for specific classes of tasks or problems and others with wider range of application;
- there is not a unique method to which might suite with entirely complexity of a real-world task of the assessment of applications for the house renovation grant system;
- the knowledge acquisition is part of the KBS development process and is included in the system's modelling activity; and
- the system description at problem space level will strongly guide the knowledge acquisition.

APPENDIX 7

case report form FROM PAST GRANT APPLICATIONS	UNIVERSITY OF SALFORD DEPARTMENT OF SURVEYING

CASE no.......... DATE........./........../.......... Progress..
Area...

1 Date property built ?

date ..../................./.................

2 Was the property

a) Built more than 10 years ago ?

Yes	
No	

b) Provide by conversion more than 10 years ago ?

Yes	
No	

3 Has the applicant made any previous application (s) for grant ?

Yes	
No	

4 If yes to the question 2 , describe the grant history regarding the property and the applicant .

____
____
____
____

5 Appication type?

please tick

Individual		Priority		First applic.	
Company		Non-priority		Sucess.app.	

6 Type of property and number of bedrooms?

	Type	Number of Bedrooms				
		1	2	3	4	5
Terraced						
Mid-Terraced Dwelling						
End Terraced Dwelling						
Semi-Detached Dwelling						
Detached Dwelling						
Flat(s)						
Cottage						
HMO						
Common Parts of a Building						

7 Did the works carried out involve converting the property to provide one or more dwellings ?

Yes	
No	

8 Was the property vacant ?

Yes	
No	

9 Was the property the main residence of the applicant(s) ?

Yes	
No	

10 If no to the question 8 , what was the purpose of the property ?

____

11 Was the property inside or intended to be in an area based scheme prior the commencement of works?

	Please tick		Please tick
Inside a group repair scheme		Intend.to be included in a group repair within 12 M	
Inside an area renewal area		Intended to be included in a renewal area	
Inside a clearance area		Intended to be included in a clearance area	
Outside any area based scheme			

12 Was the property fit/unfit for human habitation prior the commencement of works ?

	Unfit	Fit
Please tick		

13 If unfit to the question 11, what was the reason of failure ?

Please tick

	Yes	No	Unfit
Structurally stable			
Free from serious disrepair			
Free from dampness prejudicial to the health of the occupants			
Has adequate provision for lighting			
Has adequate provision for heating			
Has adequate provision for ventilation			
Has adequate piped supply of wholesome water			
There are satisfa.facilities for the prep.and cooking of food, including a sink			
Has a suitable located water-closet for the exclusive use of the occupants			
Has a suita.fixed bath or shower&wash-hand with satisfa.supply of hot /cold water			
Has effective system for draining of foul , waste and surface water			

14 Describe the condition of the property prior the commencement of works?

____

____

____

____

____

____

____

____

____

____

____

____

Condition of the property

15 What were the purposes of the relevant works ?

	Please tick		Please tick
To comply with a statutory notice		To enable to comply with certain spec.	
To make the property fit for human habitation		To repairs to prop.incl in clearance area or int. to be includ.within 12M	
To meet the ess.adapt. & facilit. needs of disabled occu.			
To put the property into reasonable repair		Rep.,improv. or adapt. to elderly app.	
To provide the dw.by the conversion of a house or build.		Adapt to a elderly occup not own/ten.	
Provision or improvement of thermal insulation		To improve above the fitness standard	
To provide adequate facilities for space heating		To the common parts of a building	
To provide satisfactory internal arrangements		To an HMO	

16 Describe the works required by the applicant (relevant works) ?

First Estimate - Total Cost £	
Second Estimate - Total Cost £	

17 Describe the eligible works defined by the Local Housing Authority ?

Schedule of Works	Cost £

18 Total amount of expenses determined in respect to the eligible works?

Total amount of expenses £ ____________________

Amount relating to the execution of eligible works £ ____________________

The amount relating to preliminar or ancillary services and charges £ ____________________

19 Applicant type ?

	Please tick		Please tick
Owner Occupier			
Landlord			
Local Authority Tenant			
Private Tenant Requi.to Carry Out Repairs			
Private Tenant Not Req.to Carry Out Rep.			
Housing Association Tenant			
Prospective Purchaser			
Participating Landlord			
Long Leaseholder			

20 What type of certificate has been provided ?

	please tick		please tick
Owner-occupation certificate			
Certificate of intended letting			
Tenant's certificate			

21 Did the applicant(s) or partner get any of the benefits listed below

	Please tick		Please tick
Housing Benefit		Unemployment Benefit	
Community Charge Benefit		Sickness Benefit	
Income Support Benefit		Disability Living Allowance	
Family Credit		Statutory Sick Pay	
Disability Working Allowance		Invalidity Benefit	
Council Tax Benefit		Severe Disablement Allowance	

22 Type of assistance applied by the applicant ?

	please tick		please tick
Renovation Grant			
Minor Works Assistance			
Common Parts Grant			
Disabled Facilities Grant			

23 Outcome of the application?

	please tick
Granted	
Nill Grant	
No Decision	
Other ____________	

24 If granted, what type of grant(s)

	please tick		please tick
Renovation Grant			
Minor Works Assistance			
Common Parts Grant			
Disabled Facilities Grant			

25 Was a mandatory or discretionary grant ?

Please tick

Mandatory	
Discretionary	

26 Amount of grant paid ?

£ ______________________

27 Amount by which the grant was reduced in accordance with section 109 or 110 of the Housing Act 1989 ?

£ ______________________

28 Amount of non-granted work ?

£ ______________________

29 Special needs of applicant/occupants ?

30 Local circunstances ?

31 Merits of the enquire ?

32 Planning issues

33 Relation with other adjoining properties

34 Results of the economic appraisal of alternative options

Renovation

Closing

Demolition

35 Proposals for the future of the area in which the property is situated

36 Needs for particular type of housing in the area

37 Effects on the community in the area

38 Effects on the environment in the area

39 Conditions of implementation

40 Additional information

APPENDIX 8

FITNESS STANDARD GUIDANCE

The guidance notes specify for each requirement of the fitness standard that a dwelling-house is considered unfit when- Guidance Notes (Department of the Environment (1990a):

Requirement: Repair

A dwelling-house is unfit for human habitation if, in the LA's view, it is in serious disrepair and for that reason it is not reasonably suitable for occupation. In deciding whether or not a dwelling-house is unfit, it should be considered whether the dwelling-house is currently free from items of disrepair which, either individually or due to their combined effect, are so severe and/or extensive that they present a risk to health and safety, or cause serious inconvenience to any occupants. Serious disrepair may be due to the severity of one item of disrepair, or due to the combined effect of two or more items. A multiplicity of items, none of which, by themselves, would be sufficiently serious to provide grounds for unfitness, may well constitute serious disrepair when combined.

Requirement: Structural stability

A dwelling-house is unfit for human habitation if in the LA's view, it is structurally unstable, and for that reason it is not reasonably suitable for occupation. In deciding whether or not a dwelling-house is unfit, it should be considered whether the dwelling-house: i) is currently able to withstand the combined dead, imposed and wind loads to which it is likely to be subjected in the ordinary course of events and when used for the purposes for which it is intended, and normal ground movement of the sub-soil caused by swelling, shrinkage, or freezing; ii) and is free from ongoing movement and the probability of movement which constitutes a threat to any occupants.

Structural stability is concerned with the basic integrity of the building, that is the stability of the basic structure or major parts of the structure down to such elements as chimneys, parapets, windows, arches and lintels. The stability of non-structural elements or of small parts of the structure which do not threaten the building's basic integrity, such as isolated areas of spalling brickwork, slipped copings or rotten floor boards, may be considered more relevant to the repair requirements.

Requirement: Dampness

A dwelling-house is unfit for human habitation if, in the LA's view, it suffers from dampness prejudicial to the health of any occupant, and for that reason it is not reasonably suitable for occupation. In deciding whether or not a dwelling-house is unfit, it should be considered whether the dwelling-house is free from occurrence of: i) rising and penetrating damp; and ii) persistent condensation and mould growth. It should be considered whether the dampness is attributable to the physical characteristics or condition of the building and whether, either directly or through its action on the structural fabric, it may be prejudicial to the health of any occupants. Dampness in a dwelling can arise from any one of the three principal causes: i) capillary attraction of ground water into the structure in contact with the ground, i.e., rising damp in the floor and walls; ii) penetration of the fabric or its joints by rainwater or melt water from standing snow; and iii) condensation.

Requirement: Ventilation

A dwelling-house is unfit for human habitation if, in the LA's view, it has inadequate provision for ventilation, and for that reason it is not reasonably suitable for occupation. In deciding whether or not a dwelling-house is unfit, it should be considered whether the dwelling-house currently has means of ventilation which, under normal conditions, are capable of restricting the accumulation of moisture and pollutants.

Requirement: Heating

A dwelling-house is unfit for human habitation if, in the LA's view, it has inadequate provision for heating, and for that reason it is not reasonably suitable for occupation. In deciding whether or not a dwelling-house is unfit, it should be considered whether the dwelling-house currently has: i) for heating a main living room, provision for fixed heating capable of efficiently maintaining the room generally at a temperature of 18 dg C or more when the outside temperature is -1 dg C; and ii) for other main habitable rooms, provision for heating capable of maintaining an equivalent temperature of 16 dg C or more. Additional should be considered whether the construction and condition of the dwelling-house: i) prevents excessive heat losses; and ii) whether the overall level of provision for heating, when combined with adequate ventilation, is sufficient to prevent both condensation and mould growth prejudicial to health.

Requirement: Lighting

A dwelling-house is unfit for human habitation if, in the LA's view, it has inadequate provision for lighting, and for that reason it is not reasonably suitable for occupation. In deciding whether or not a dwelling-house is unfit, it should be considered whether the dwelling-house currently has provision for sufficient natural lighting in habitable rooms to enable the normal activities of a household to be carried out, safely and conveniently, without the use of artificial light during normal daytime conditions. It should also be considered whether the dwelling-house has provision for sufficient artificial lighting in all habitable rooms, kitchens, bathrooms, w.c. and circulation spaces, to enable the normal activities of a household to be carried out safely.

Requirement: Water supply

A dwelling-house is unfit for human habitation if, in the LA's view, it has inadequate piped supply of wholesome water, and for that reason it is not reasonably suitable for occupation. In deciding whether or not a dwelling-house is unfit, it should be considered whether the dwelling-house: i) is currently connected to a main supply that is wholesome; ii) has, normally, a continuous and adequate rate of supply; iii) has piping designed, installed and in condition so as not to contaminate the supply; and iv) has an outlet conveniently located above the kitchen sink.

Requirement: Facilities for the preparation and cooking of food

A dwelling-house is unfit for human habitation if, in the LA's view, it lacks satisfactory facilities for the preparation and cooking of food, and for that reason it is not reasonably suitable for occupation. In deciding whether or not a dwelling-house is unfit, it should be considered whether the dwelling-house currently has: i) a sink designed and installed so as not to be prejudicial to health and fitted with satisfactory supplies of hot water and cold drinking water; ii) provision for an adequate work surface(s) for the preparation and cooking of food; iii) provision for a gas or electric cooker or, failing this, a suitable fixed solid fuel or oil fired cooking appliance.

Requirement: Water closet, washbasin and bath or shower

A dwelling-house is unfit for human habitation if, in the LA's view, it lacks for the exclusive use of any occupants, a suitably located water closet or a suitably located fixed bath or shower and wash-hand basin, each with a satisfactory supply of hot and cold water, and for that reason it is not reasonably suitable for occupation. In deciding whether or not a dwelling-house is unfit, it should be considered whether the dwelling-house currently has: i) a water closet, for the exclusive use of any occupants, designed, installed and suitably located inside so as to be readily accessible and not prejudicial to health; and ii) a wash-hand basin and fixed bath or shower, for the exclusive use of any occupants, designed, installed and suitably located inside so as to be readily accessible and not prejudicial to health, and fitted with satisfactory supplies of hot and cold water.

Requirement: Drainage of foul, waste and surface water

A dwelling-house is unfit for human habitation if, in the LA's view, it lacks an effective system for the draining of foul, waste and surface water, and for that reason it is not reasonably suitable for occupation. In deciding whether or not a dwelling-house is unfit, it should be considered whether the dwelling-house currently has an effective system, both above and below ground, for the draining of foul, waste and surface water.

APPENDIX 9

Predictive Validation

Renovation Grant ( )
Disabled Facilities Grant ( )
Minor Works Assistance ( )

Test Cases : Past Applications

Code No.:_______________

1. Performance and functional features

Required performance - The system should represent the knowledge and expertise required to assess the applications for renovation grants within the limits of the research
Actual performance :   ***Highly appropriate ( ) Appropriate ( ) Generally appropriate but need some refinement ( ) Less appropriate ( ) Not appropriate ( ) Other*** _______________   ____________________   ____________________

Required performance - The case-based reasoning should allow the system to propose quickly reliable solutions within the limits of the research
Actual performance :   ***Very fast ( ) Fast ( ) Reasonably fast ( ) Equivalent to other means ( ) Slow ( ) Very slow ( )***   ____________________   ____________________

Required performance - The case-based reasoning should allow the system to derive solutions to new application from past applications properly indexed
Actual performance :   ***Very satisfactory ( ) Satisfactory ( ) Reasonably satisfactory( ) Unsatisfactory( ) Very Unsatisfactory ( )***   ____________________   ____________________

Required performance - The case-based reasoning should provide and adequate mean for indexing knowledge and expertise acquired from the past applications which can be ready available for new applications
Actual performance :   ***Very adequate( ) Adequate( ) Reasonably adequate( ) Inadequate ( ) Very inadequate( )***   ____________________   ____________________

Predictive Validation

Renovation Grant ( )
Disabled Facilities Grant ( )
Minor Works Assistance ( )

Test Cases : Past Applications

Code No.:________________

Very adequate( ) Adequate( ) Reasonably adequate( ) Inadequate ( ) Very inadequate( )

Required performance - The case-based reasoning should provide and adequate mean to maintain the case database of experiences acquired from past applications

Actual performance :

Very adequate( ) Adequate( ) Reasonably adequate( ) Inadequate ( ) Very inadequate( )

Required performance - The case-based reasoning should allow the system to learn from experience by accumulating and integrating new experiences in its case database and indexing them appropriately

Actual performance :

Very acceptable ( ) Acceptable ( ) Reasonably acceptable ( ) Unacceptable ( ) Very unacceptable ( )

Required performance - The case-based reasoning should allow the system when fail to retrieve a past solution to construct a new solution using old experience

Actual performance :

Very acceptable ( ) Acceptable ( ) Reasonably acceptable ( ) Unacceptable ( ) Very unacceptable ( )

Required performance - The case-based reasoning should provide an adequate framework , integrating other reasoning methods , for a system designed as decision aid to the assessment of applications for renovation grants

Actual performance :

Very adequate( ) Adequate( ) Reasonably adequate( ) Inadequate ( ) Very inadequate( )

Predictive Validation

Renovation Grant ( )
Disabled Facilities Grant ( )
Minor Works Assistance ( )

Test Cases : Past Applications

Code No.:______________

Required performance - The system should be able to perform the preliminary assessment a new application and to recommend a decision

Actual performance :

Very acceptable ( ) Acceptable ( ) Reasonably acceptable ( ) Unacceptable ( ) Very unacceptable ( )

Required performance - The system should be able to perform the fitness assessment of a property relating to a new application and to recommend a decision about its fitness

Actual performance :

Very acceptable ( ) Acceptable ( ) Reasonably acceptable ( ) Unacceptable ( ) Very unacceptable ( )

Required performance - The system should be able to perform the disabled facilities assessment of a property relating to a new application and to recommend a decision about its adaptations

Actual performance :

Very acceptable ( ) Acceptable ( ) Reasonably acceptable ( ) Unacceptable ( ) Very unacceptable ( )

Required performance - The system should be able to perform the means test for a new application within the limits of the research

Actual performance :

Very acceptable ( ) Acceptable ( ) Reasonably acceptable ( ) Unacceptable ( ) Very unacceptable ( )

Predictive Validation

Renovation Grant ( )
Disabled Facilities Grant ( )
Minor Works Assistance ( )

Test Cases : Past Applications

Code No.:______________

Required performance* - *The system should be able to recommend the schedule of works, including its costs, to make fit an unfit property or for the required adaptation works for a disabled occupant or for required minor assistance

Actual performance :

Very acceptable ( ) Acceptable ( ) Reasonably acceptable ( ) Unacceptable ( ) Very unacceptable ( )

Required performance* - *In case of an application for renovation grants , the system should be able to perform the economic analysis of the proposed renovation action

Actual performance :

Very acceptable ( ) Acceptable ( ) Reasonably acceptable ( ) Unacceptable ( ) Very unacceptable ( )

Required performance* - *The system should be able to determine the degree of eligibility of a new application

Actual performance :

Very acceptable ( ) Acceptable ( ) Reasonably acceptable ( ) Unacceptable ( ) Very unacceptable ( )

Required performance* - *The system should be able to recommend the amount of grant to be award to a new application

Actual performance :

Very acceptable ( ) Acceptable ( ) Reasonably acceptable ( ) Unacceptable ( ) Very unacceptable ( )

Predictive Validation

Renovation Grant ( )
Disabled Facilities Grant ( )
Minor Works Assistance ( )

Test Cases : Past Applications

Code No.:_______________

Required performance - The system should be able to recommend and report a decision on a new application
Actual performance :   *Very acceptable ( ) Acceptable ( ) Reasonably acceptable ( ) Unacceptable ( ) Very unacceptable ( )*
Required performance - The system should be able to explain the recommended decisions provided during the consultation
Actual performance :   *Very acceptable ( ) Acceptable ( ) Reasonably acceptable ( ) Unacceptable ( ) Very unacceptable ( )*
Required performance - The system should be able to support the organizational knowledge learning/creation
Actual performance :   *Very acceptable ( ) Acceptable ( ) Reasonably acceptable ( ) Unacceptable ( ) Very unacceptable ( )*

2. Validation by criteria

Required criteria : Accuracy and correctness- success rate against known results
Actual performance :   *Equal to past result ( ) Very close to past result ( ) Close to past result ( ) Far from past result ( ) Very far from past result ( )*

Predictive Validation

Renovation Grant ( )
Disabled Facilities Grant ( )
Minor Works Assistance ( )

Test Cases : Past Applications

Code No.:______________

Required criteria : Quality of solution - against known results

Actual performance :

Much better than past result ( ) Somewhat better than past result ( ) Equivalent to past result ( ) Somewhat poor than past result ( ) Much poor than past result ( )

Required criteria : Generality - against known results

Actual performance :

Very suitable to work in the context ( ) Suitable to work in the context ( ) Reasonable suitable to work in the context ( ) Not suitable to work in the context ( )

Required criteria : Usefulness - against known results

Actual performance :

Contains all necessary and adequate parameters and relationships for use in the present problem ( ) Reasonable contains all necessary and adequate parameters and relationships for use in the present problem ( ) Does not contains all necessary and adequate parameters and relationships for use in the present problem ( )

Required criteria : Adaptability - against known results

Actual performance :

Very easily adaptable ( )Easily adaptable ( ) Reasonable adaptable ( ) Not adaptable ( )

Predictive Validation

Renovation Grant ( )
Disabled Facilities Grant ( )
Minor Works Assistance ( )

Test Cases : Past Applications

Code No.:_______________

3. Validation of Results - Intermediate and final

Required criteria : Accuracy - against known results

Actual performance :

	Results	
<u>***Task***</u>	***System***	***Expert***
a) Preliminary assessment	____________________	____________________
b) Fitness assessment		
Fitness	____________________	____________________
Reasons for unfitness	____________________	____________________

Actual performance :

	Results	
	System	***Expert***
c) Means test		
Applicable amount	____________________	____________________
Eligible Income	____________________	____________________
Contrib. to grant	____________________	____________________

Required criteria : Accuracy - against known results

d) Schedule of Works	____________________	____________________
e) Cost of works	____________________	____________________
f)Economic		

Predictive Validation

Renovation Grant ( )
Disabled Facilities Grant ( )
Minor Works Assistance ( )

Test Cases : Past Applications

Code No.:_______________

analysis	_______________	_______________
g)Amount of grant	_______________	_______________
h) Non granted	_______________	_______________
i) Degree of eligibility	_______________	_______________
j) Final Decision	_______________	_______________

Required criteria : Time to solution - against known results

Actual performance :

	Results	
	System	***Expert***
Time to run all system (minutes)	_______________	_______________

Field Validation

Renovation Grant ( )
Disabled Facilities Grant ( )
Minor Works Assistance ( )

COMMENTS SHEET
Test Cases : New Applications

Code No.:______________

1. System effectiveness

Required performance - The case-based reasoning should allow the system to propose quickly reliable solutions within the limits of the research

Actual performance :
Very fast ( ) Fast ( ) Reasonably fast ( ) Equivalent to other means ( ) Slow ( ) Very slow ( )

Required performance - The case-based reasoning should allow the system to derive accurate and correct solutions to new application from past applications

Actual performance :
Very satisfactory ( ) Satisfactory ( ) Reasonable satisfactory( ) Unsatisfactory( ) Very Unsatisfactory ( )

Required performance - The case-based reasoning should allow the system to learn from experience by accumulating and integrating new experiences in its case database and indexing them appropriately

Actual performance :
Very acceptable ( ) Acceptable ( ) Reasonable acceptable ( ) Unacceptable ( ) Very unacceptable ( )

Required performance - The case-based reasoning should allow the system when fail to retrieve a past solution to construct a new solution using old experience

Actual performance :
Very acceptable ( ) Acceptable ( ) Reasonable acceptable ( ) Unacceptable ( ) Very unacceptable ( )

Field Validation

Renovation Grant ( )
Disabled Facilities Grant ( )
Minor Works Assistance ( )

COMMENTS SHEET
Test Cases : New Applications

Code No.:_______________

Required performance - The system should be able to perform the preliminary assessment a new application and to recommend a decision

Actual performance :
Very acceptable ( ) Acceptable ( ) Reasonable acceptable ( ) Unacceptable ( ) Very unacceptable ( )

____

____

Required performance - The system should be able to perform the fitness assessment of a property relating to a new application and to recommend a decision about its fitness

Actual performance :
Very acceptable ( ) Acceptable ( ) Reasonable acceptable ( ) Unacceptable ( ) Very unacceptable ( )

____

____

Required performance - The system should be able to perform the fitness assessment of a property relating to a new application and to recommend a decision about its fitness

Actual performance :
Very acceptable ( ) Acceptable ( ) Reasonable acceptable ( ) Unacceptable ( ) Very unacceptable ( )

____

____

Required performance - The system should be able to perform the disabled facilities assessment of a property relating to a new application and to recommend a decision about its adaptations

Actual performance :
Very acceptable ( ) Acceptable ( ) Reasonable acceptable ( ) Unacceptable ( ) Very unacceptable ( )

____

____

Field Validation

Renovation Grant ( )
Disabled Facilities Grant ( )
Minor Works Assistance ( )

COMMENTS SHEET
Test Cases : New Applications

Code No.:______________

Required performance - The system should be able to perform the means test for a new application within the limits of the research
Actual performance : ***Very acceptable ( ) Acceptable ( ) Reasonable acceptable ( ) Unacceptable ( ) Very unacceptable ( )*** ______________________________ ______________________________
Required performance - The system should be able to recommend the schedule of works, including its costs, to make fit an unfit property or for the required adaptation works for a disabled occupant or for required minor assistance
Actual performance : ***Very acceptable ( ) Acceptable ( ) Reasonable acceptable ( ) Unacceptable ( ) Very unacceptable ( )*** ______________________________ ______________________________
Required performance - In case of an application for renovation grants , the system should be able to perform the economic analysis of the proposed renovation action
Actual performance : ***Very acceptable ( ) Acceptable ( ) Reasonable acceptable ( ) Unacceptable ( ) Very unacceptable ( )*** ______________________________ ______________________________
Required performance - The system should be able to determine the degree of eligibility of a new application
Actual performance : ***Very acceptable ( ) Acceptable ( ) Reasonable acceptable ( ) Unacceptable ( ) Very unacceptable ( )*** ______________________________ ______________________________

Field Validation

Renovation Grant ( )
Disabled Facilities Grant ( )
Minor Works Assistance ( )

COMMENTS SHEET
Test Cases : New Applications

Code No.:______________

Required performance - The system should be able to recommend the amount of grant to be award to a new application
Actual performance : ***Very acceptable ( ) Acceptable ( ) Reasonable acceptable ( ) Unacceptable ( ) Very unacceptable ( )***

Required performance - The system should be able to recommend and report a decision on a new application
Actual performance : ***Very acceptable ( ) Acceptable ( ) Reasonable acceptable ( ) Unacceptable ( ) Very unacceptable ( )***

Required performance - The system should be able to fulfil its purposes and facilitate use of its sub components
Actual performance : ***Very acceptable ( ) Acceptable ( ) Reasonable acceptable ( ) Unacceptable ( ) Very unacceptable ( )***

2. User effectiveness

Required performance - The system should be able to explain the recommended decisions provided during the consultation
Actual performance : ***Very acceptable ( ) Acceptable ( ) Reasonable acceptable ( ) Unacceptable ( ) Very unacceptable ( )***

Field Validation

Renovation Grant ( )
Disabled Facilities Grant ( )
Minor Works Assistance ( )

COMMENTS SHEET
Test Cases : New Applications

Code No.:________________

Required performance - The system should be able to facilitate the specification of inputs and to provide outputs(results) whose form and contents are easy to assimilate and useful
Actual performance :   ***Very acceptable ( ) Acceptable ( ) Reasonable acceptable ( ) Unacceptable ( ) Very unacceptable ( )***   ____   ____

Required performance : Time to solution
Actual performance :   ***Results***   ***System*** | ***Expert***   ***Time to run all system (minutes)*** ______________ ______________

Required performance - The system should be able to support the organizational knowledge learning/creation
Actual performance :   ***Very acceptable ( ) Acceptable ( ) Reasonably acceptable ( ) Unacceptable ( ) Very unacceptable ( )***   ____   ____

Face Validation

Questionnaire And Comments Given By Experts

1. How satisfied are you generally with the performance of the system ?

	please tick
Very satisfied	
Satisfied	
Reasonably satisfied	
Dissatisfied	
Very dissatisfied	

2. How well do you believe the system is potentially suited for the house renovation grants ?

	please tick
Very suitable	
Suitable	
Somewhat suitable	
Neutral	
Not suitable	

3. In what way(s) do you think the system can be potenlially useful ?

	please tick
Preliminary assessment	
Fitness assessment	
Disabled facilities assessment	
Schedule and cost of works	
Economic analysis	
Means test	
Degree of eligibility	
Final decision	

4. What do you think of the benefits of using the system ?

	Significant	Major	Medium	Some	Trivial
Time savings					
Cost savings					
Decision improvement					
Efficiency improvement					
Consistency on task					
Accessibility to expertise					
Training					
Standard reports					
ıderstandability increase					

5. What do you think about the system efficiency in terms of its purposes and use ?

	please tick
Very satisfied	
Satisfied	
Reasonably satisfied	
Dissatisfied	
Very dissatisfied	

6. To what extent do you think the case based reasoning framework used is suited to model the task of assessing house renovation applications ?

	please tick
Very suitable	
Suitable	
Somewhat suitable	
Neutral	
Not suitable	

7. To what extent do you think the object oriented approach used is appropropriate to implement the system in the computer ?

	please tick
Very appropriate	
Appropriate	
Generally appropriate	
Less appropriate	
Not appropriate	

8. What do you think of the indexing system used to represent the cases (past applications) ?

	please tick
Very appropriate	
Appropriate	
Generally appropriate	
Less appropriate	
Not appropriate	

9. What do you think of the retrieval mechanism used in the system ?

	please tick
Very appropriate	
Appropriate	
Generally appropriate	
Less appropriate	
Not appropriate	

10. To what extent do you think the introduction of the preliminary assessment, economic analysis and dregree of eligibility can improve the decision making process?

	please tick
Very much	
Much	
Some	
Not at all	

11. Which of the following can best describe the user interfaces(e.g., screen layout, inputs , output reports,reasoning transparency) used in the system ?

	please tick
Very appropriate	
Appropriate	
Generally appropriate	
Less appropriate	
Not appropriate	

12. To what extent do you think about the usefulness of the explanations provided by the system ?

	please tick
Very useful	
Useful	
Somewhat useful	
Neutral	
Not useful	

13. To what extent do you think the system can enhances organisational knowledge learning ?

	please tick
Very much	
Much	
Some	
Not at all	

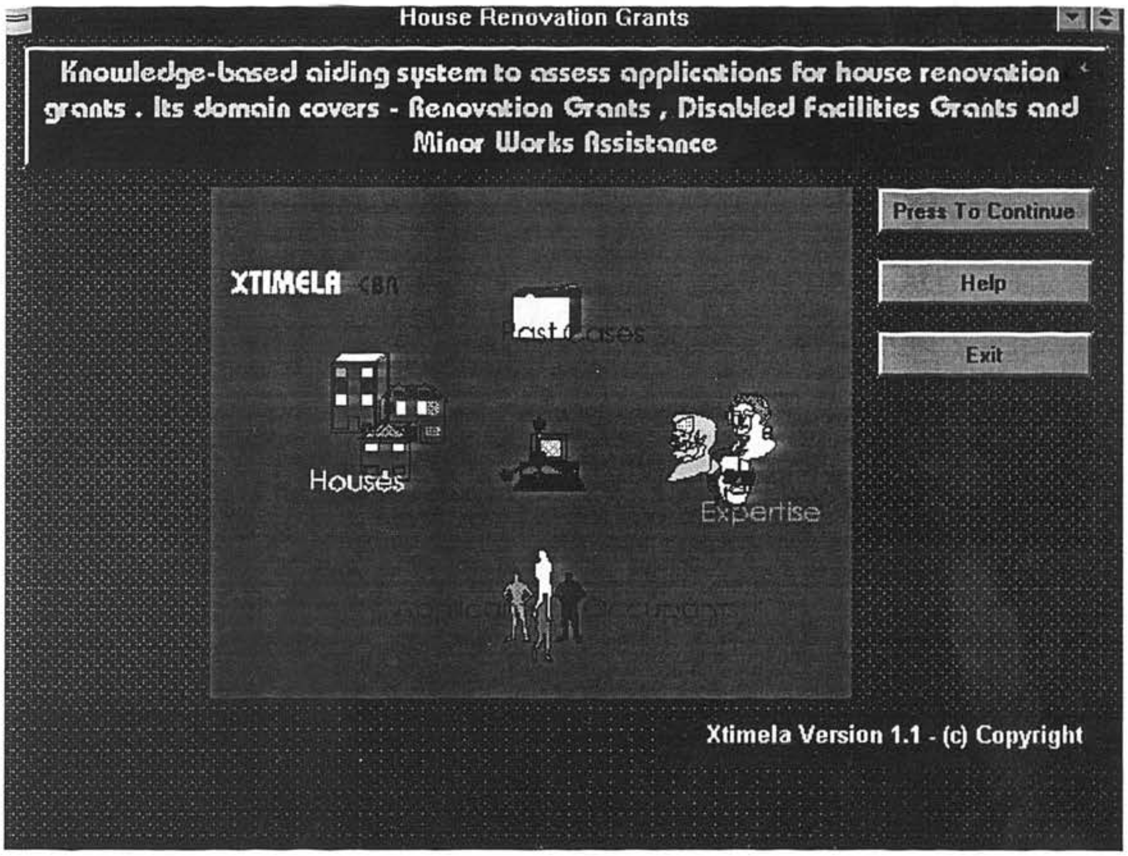
14. What do you think of the framework potentiality for a large-scale departmental-wid case-based system ?

	please tick
Very useful	
Useful	
Somewhat useful	
Neutral	
Not useful	

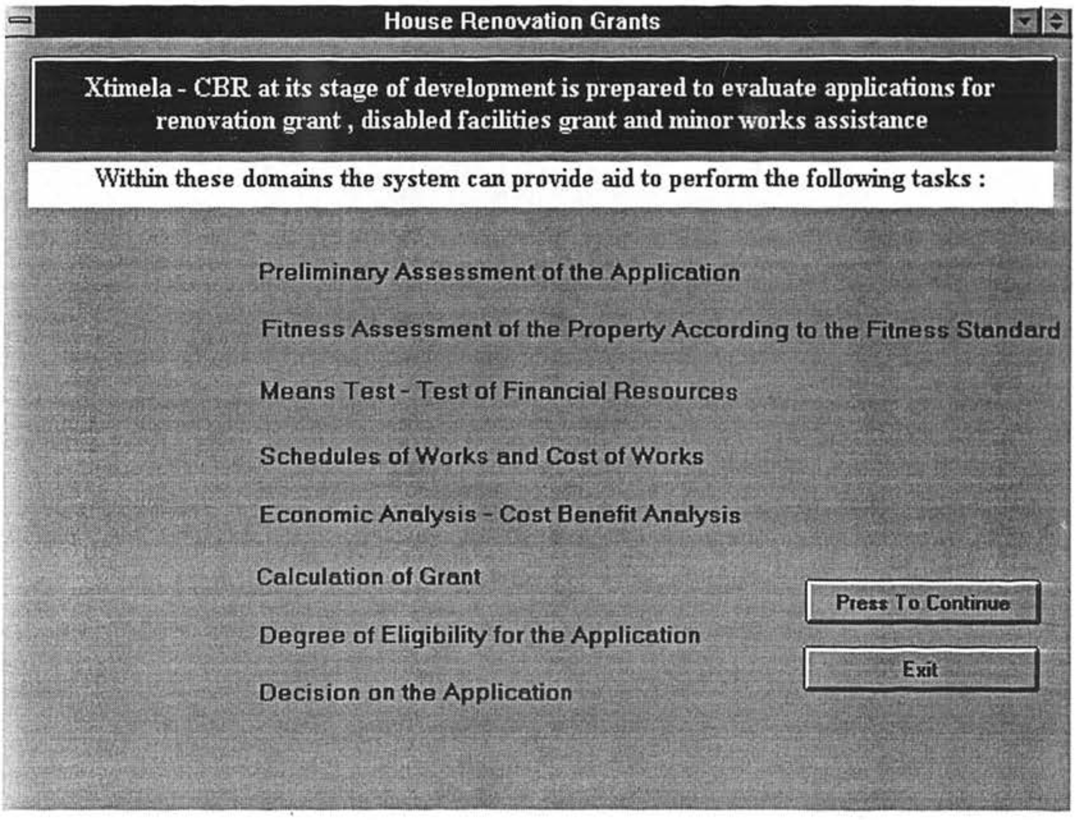
APPENDIX 10

A TYPICAL PROBLEM SOLVING SESSION FOR ASSESSING AN APPLICATION FOR A RENOVATION GRANT

10.1- INTRODUCTION

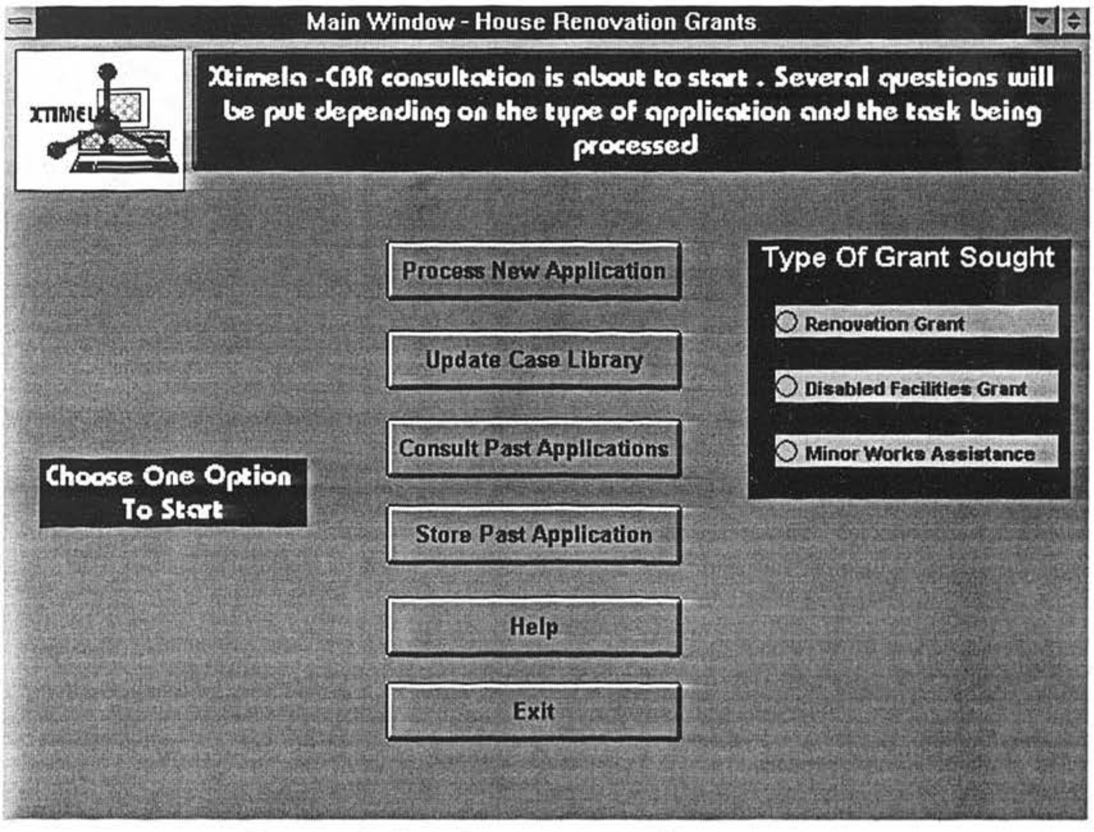


Screen 1: Introduction of the system

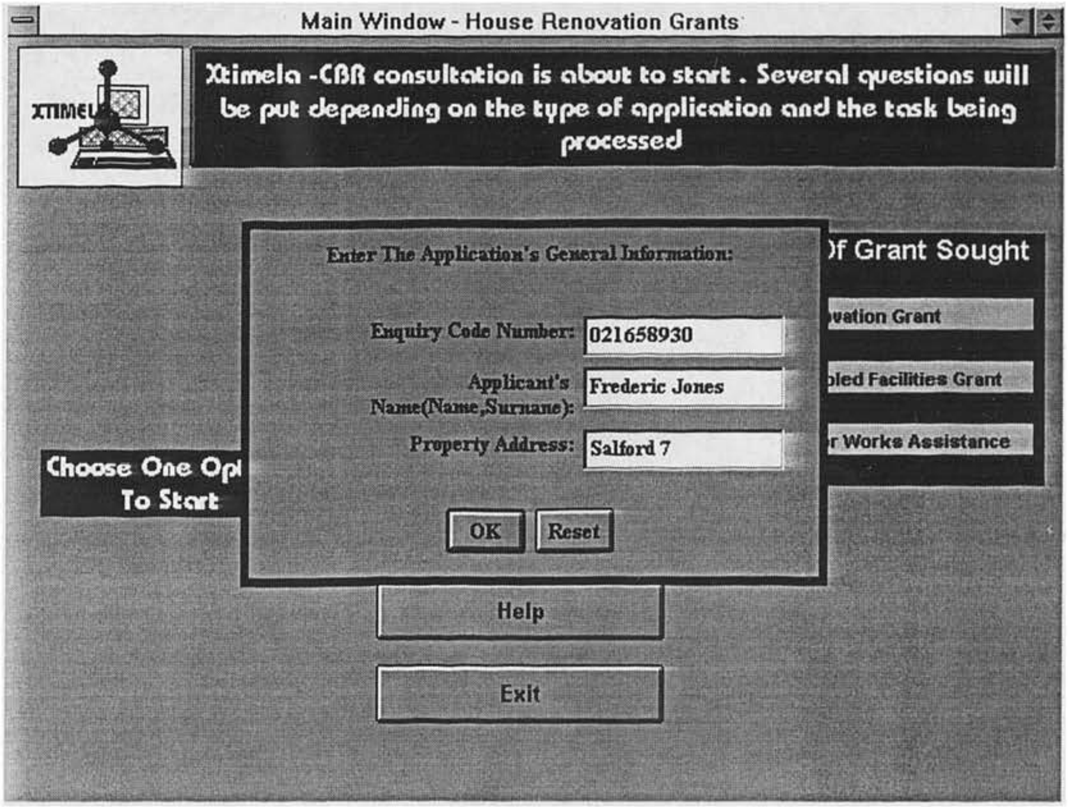


Screen 2: Main system's tasks

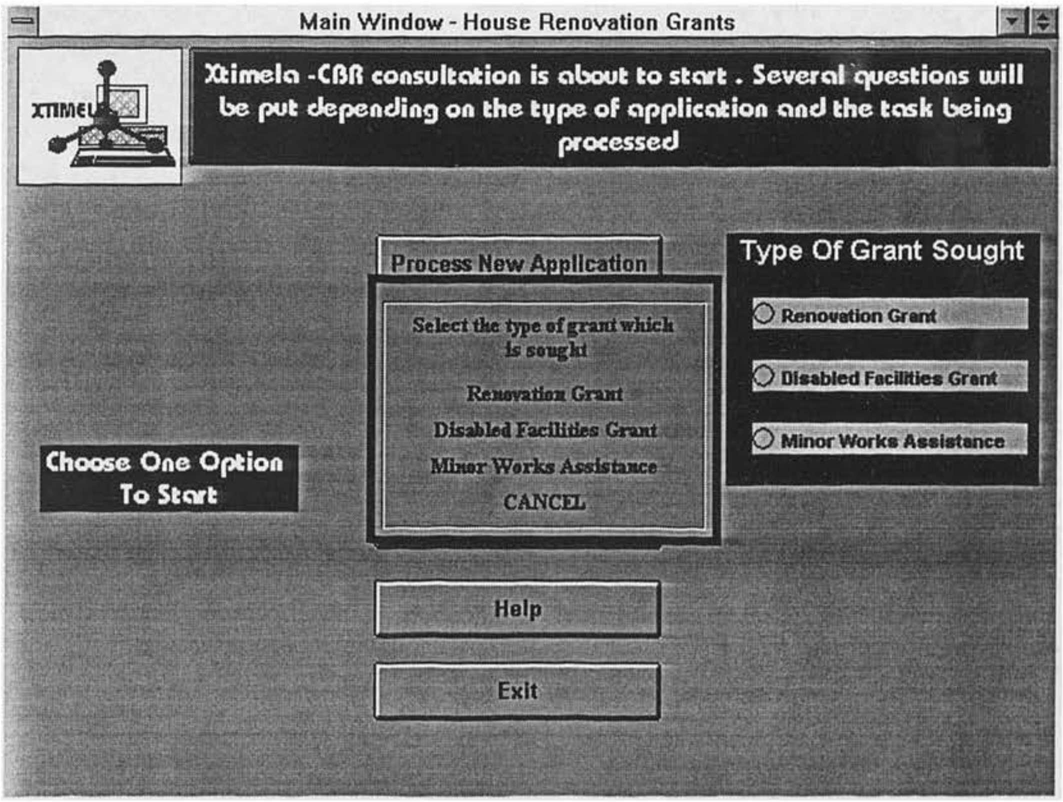
10.2- NEW APPLICATION CASE GENERATION



Screen 3: Case-base manger sub components

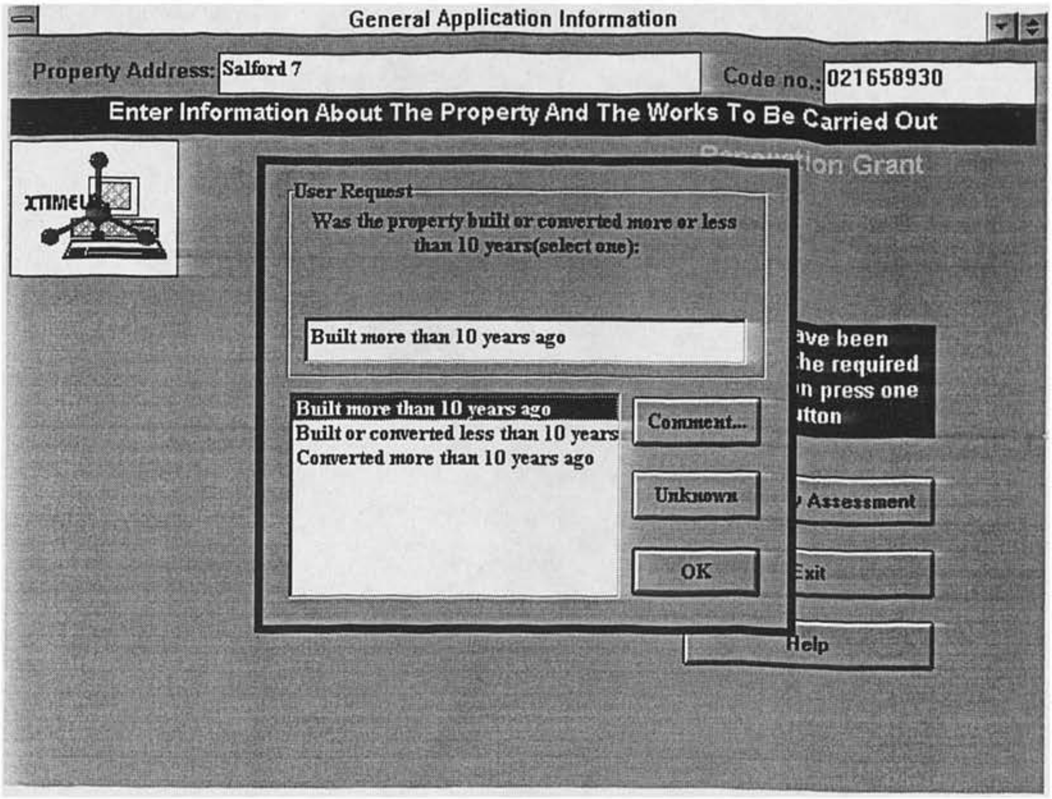


Screen 4: General data for creating a new application case

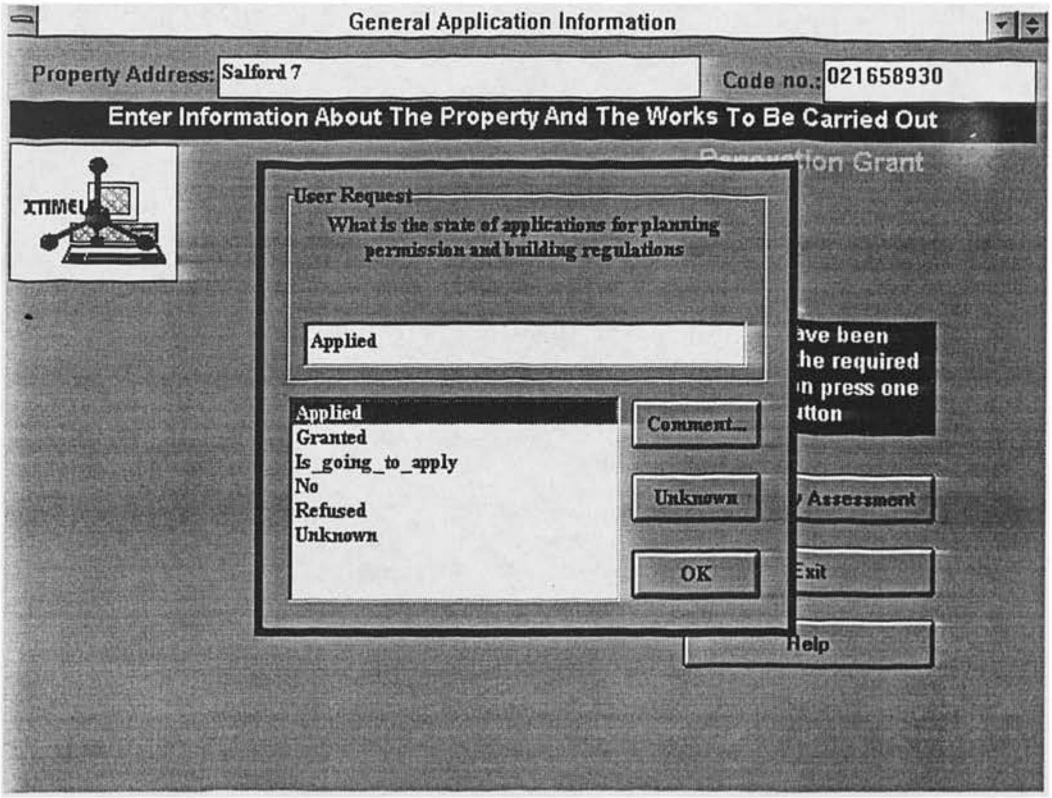


Screen 5: Input type of grant

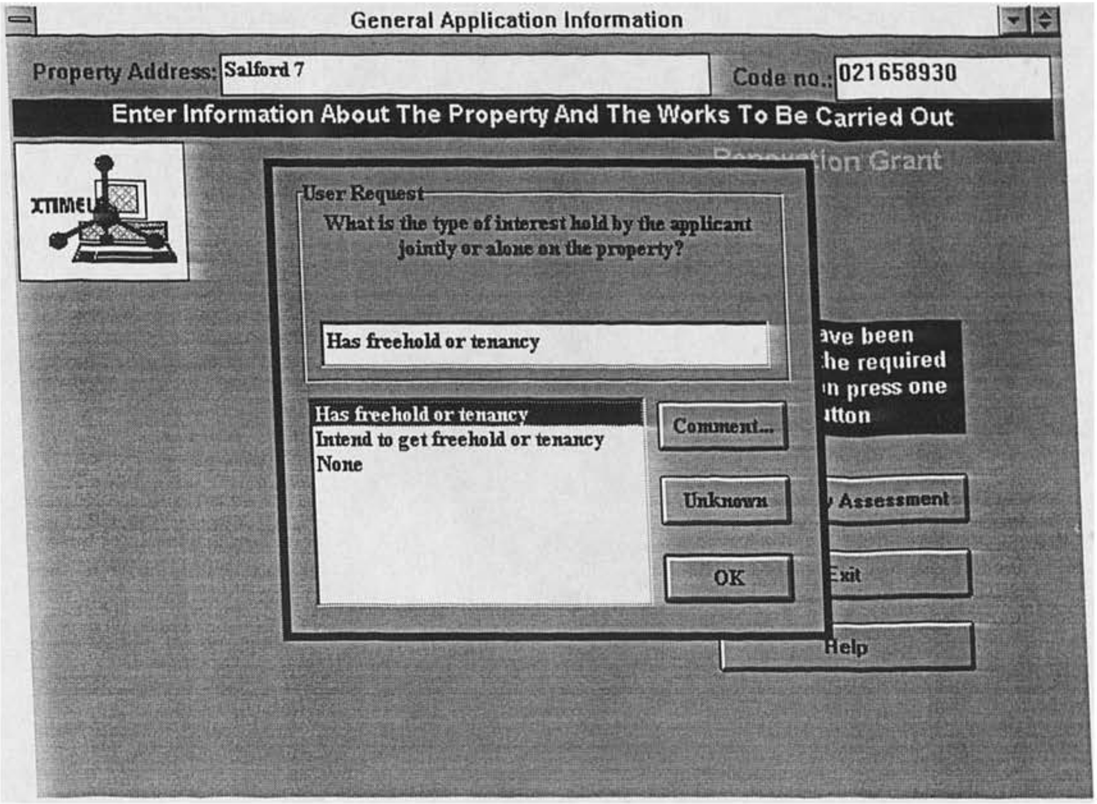
10.3- ENQUIRY ELIGIBILITY



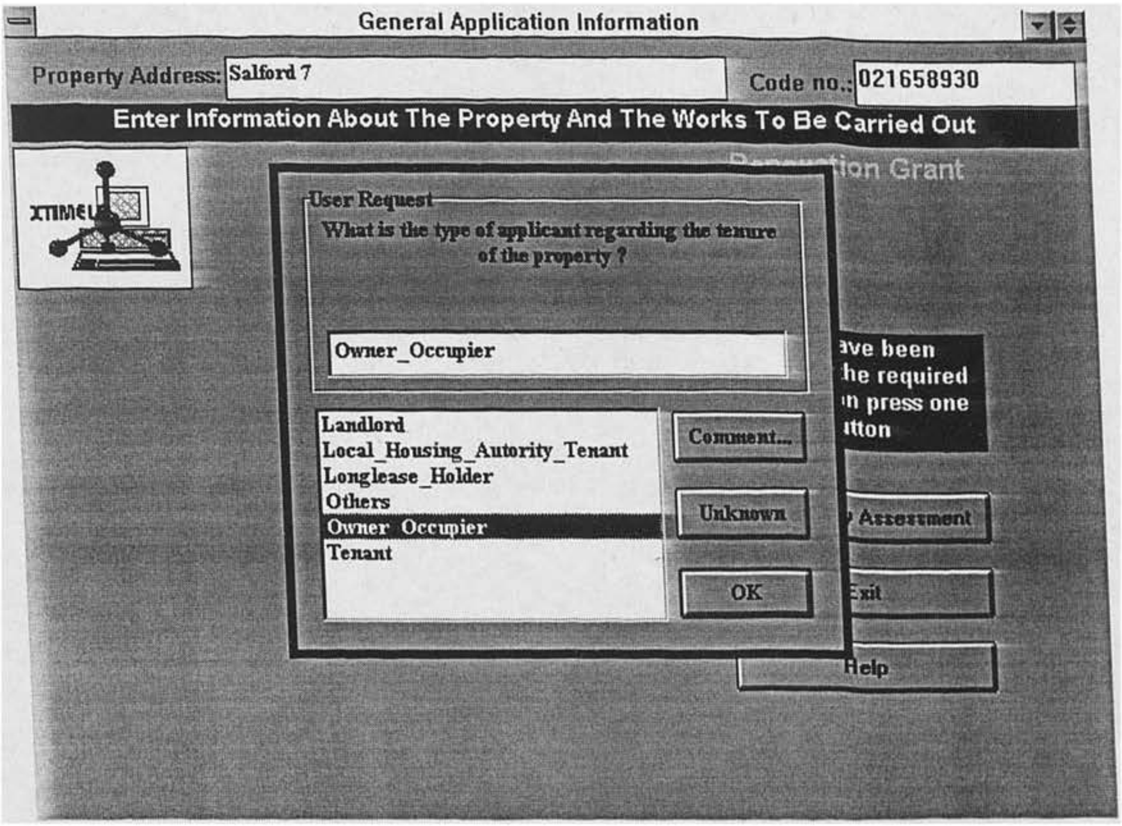
Screen 6: Input enquiry information- Age of the property



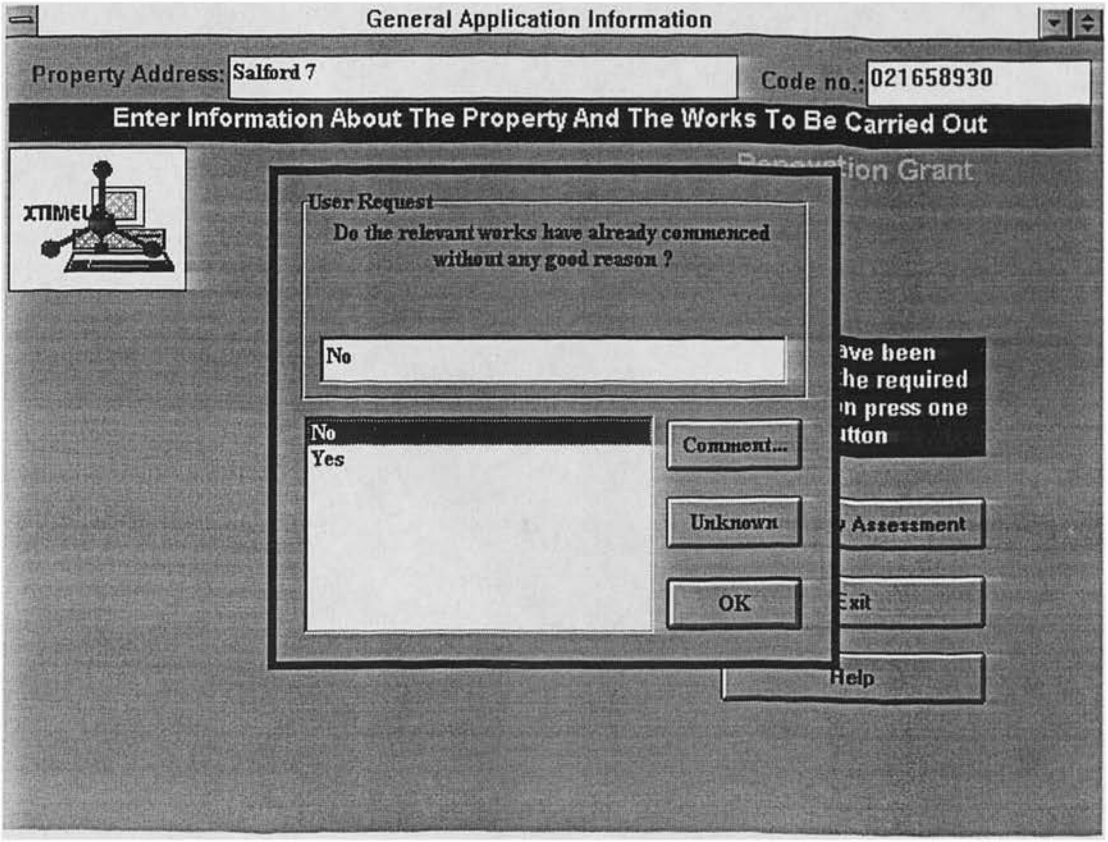
Screen 7: Input enquiry information-about planing permission and Building Regulations approval



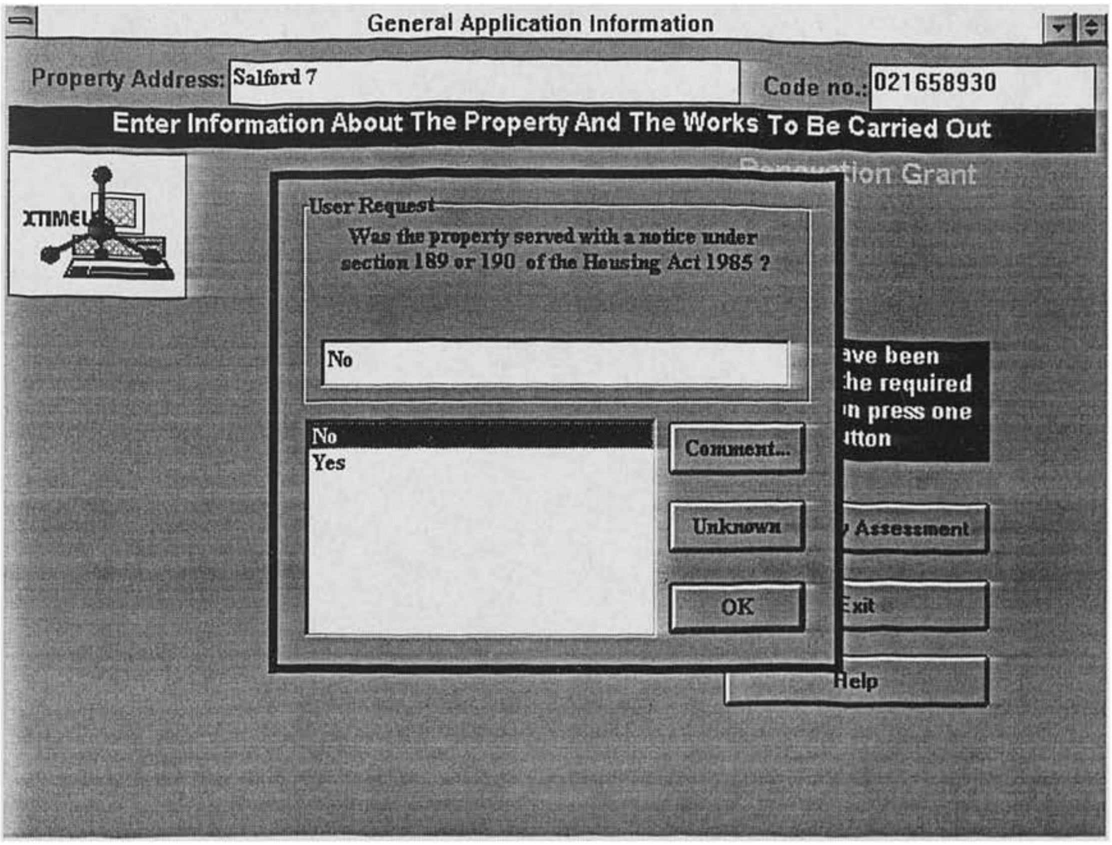
Screen 8: Input enquiry information-about the applicant's interest on the property



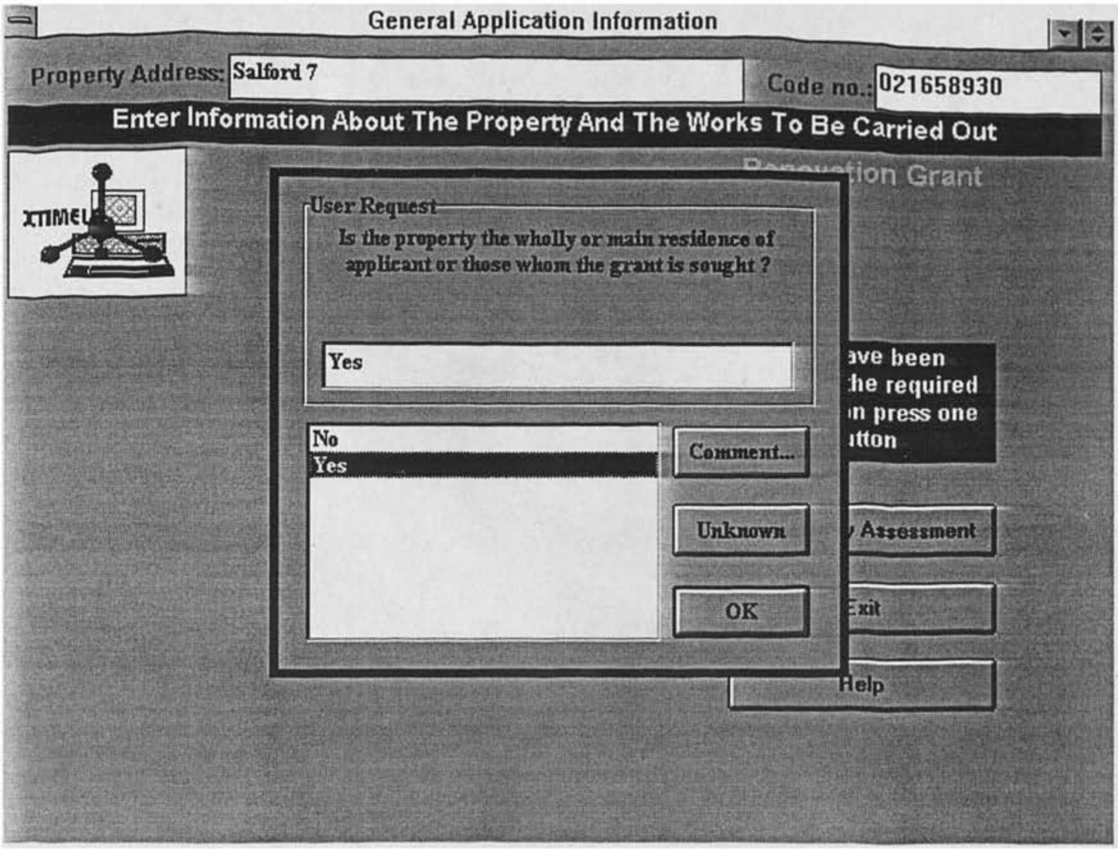
Screen 9: Input enquiry information-about the tenure of the property



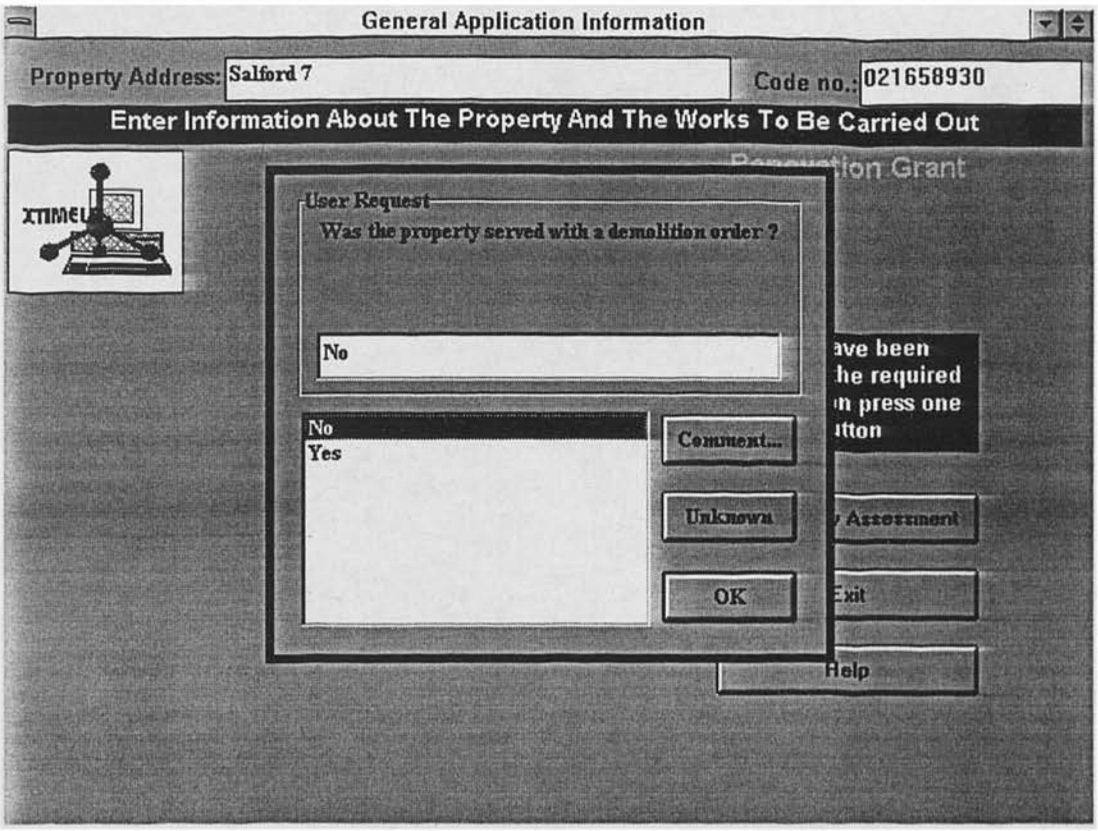
Screen 10: Input enquiry information-about the state of works



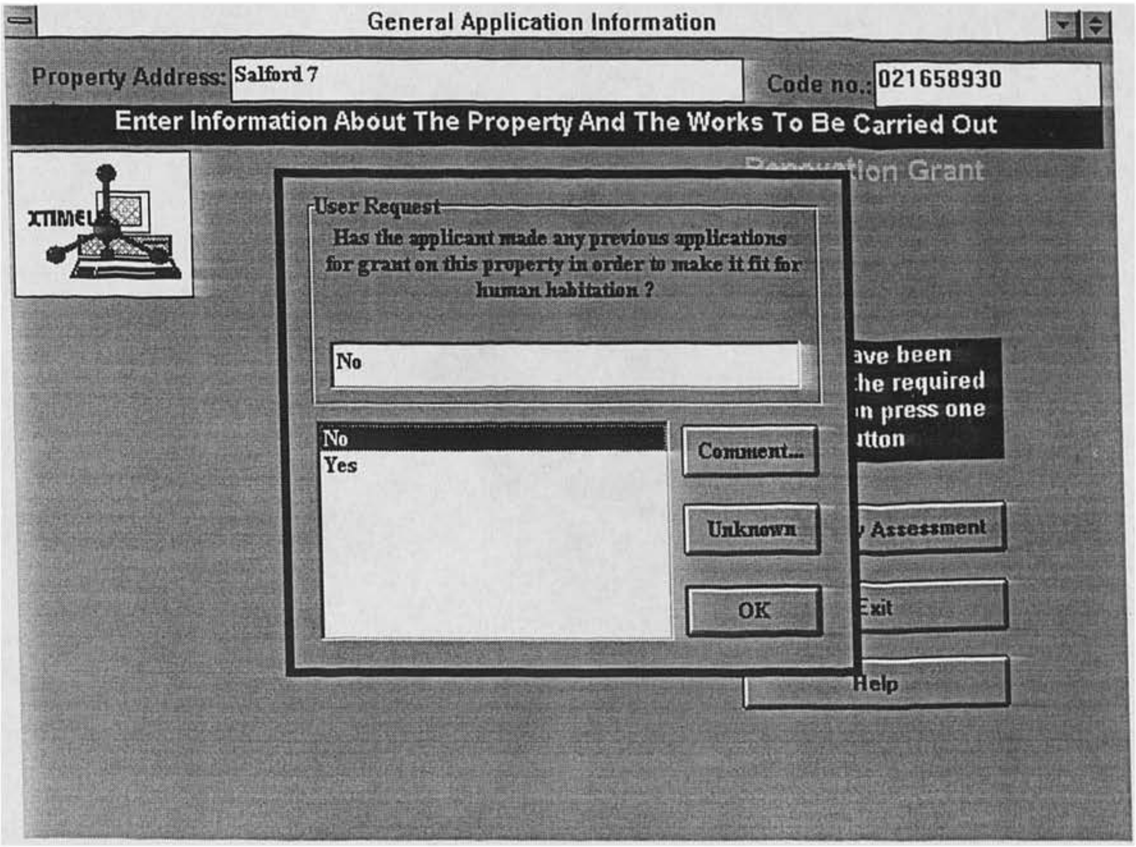
Screen 11: Input enquiry information-about the existence of any repair notice for the property



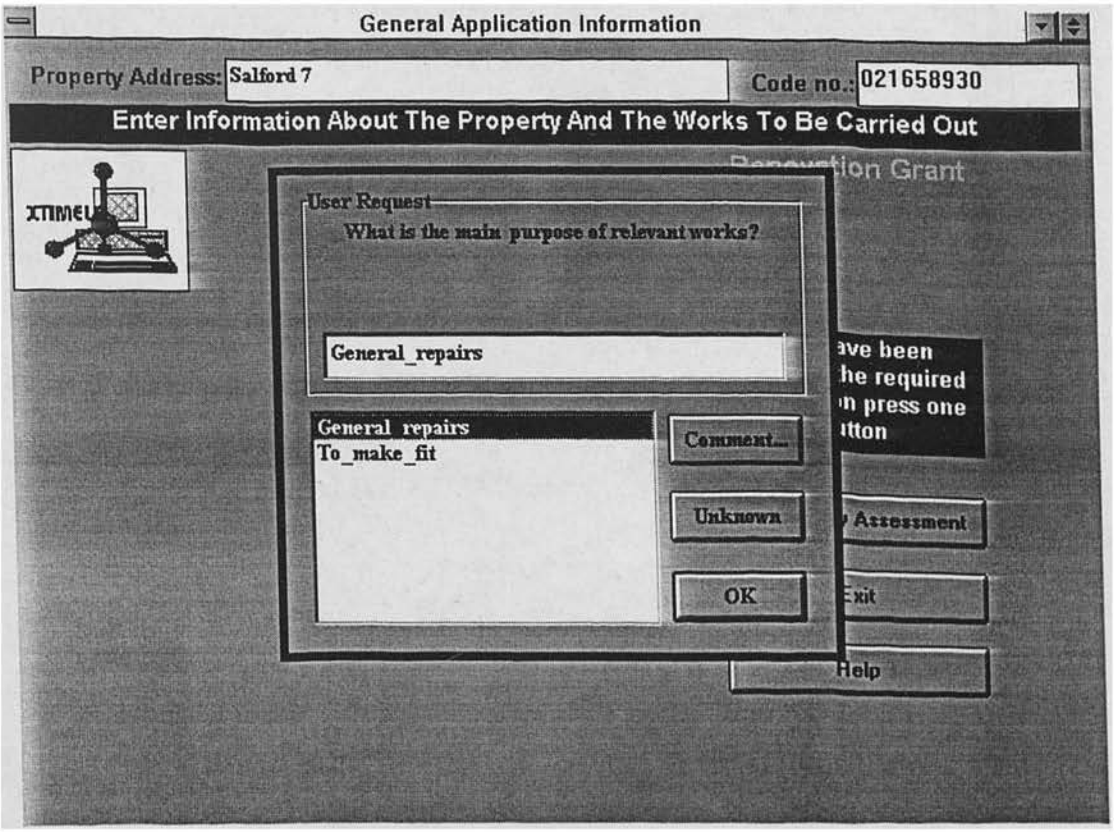
Screen 12: Input enquiry information-about the type of use of the property



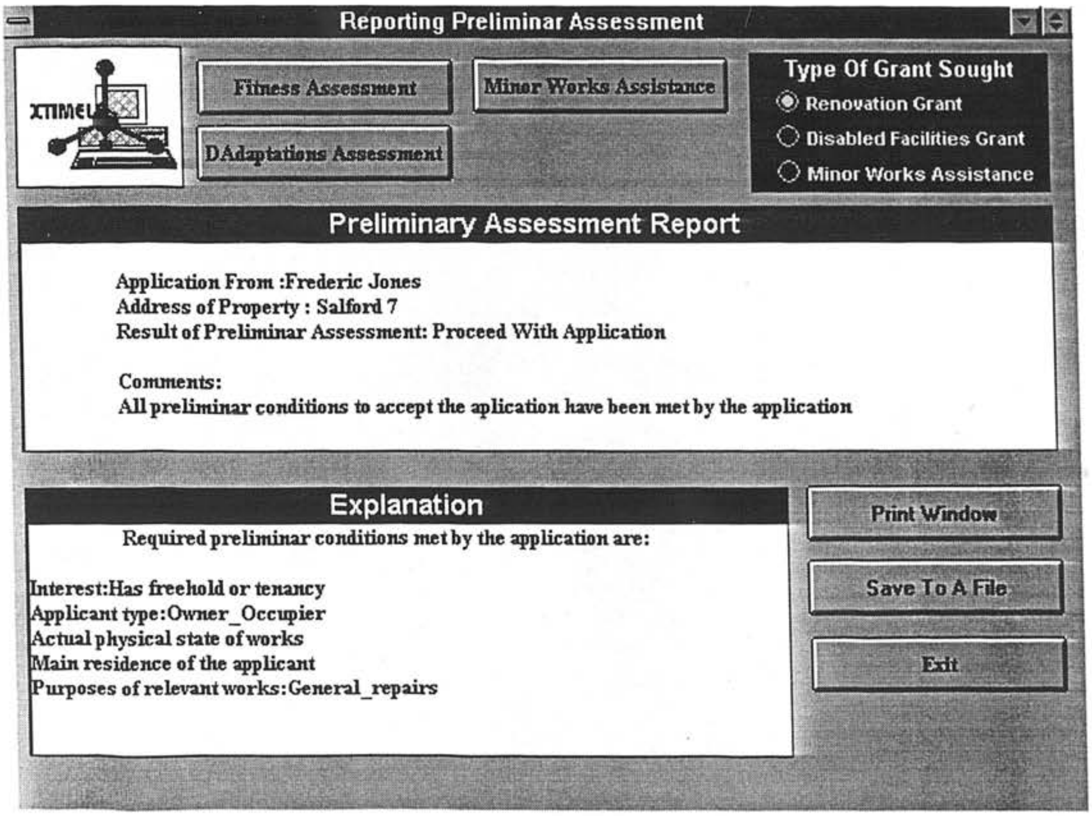
Screen 13: Input enquiry information-about the existence of any demolition order for the property



Screen 14: Input enquiry information-about the grant history



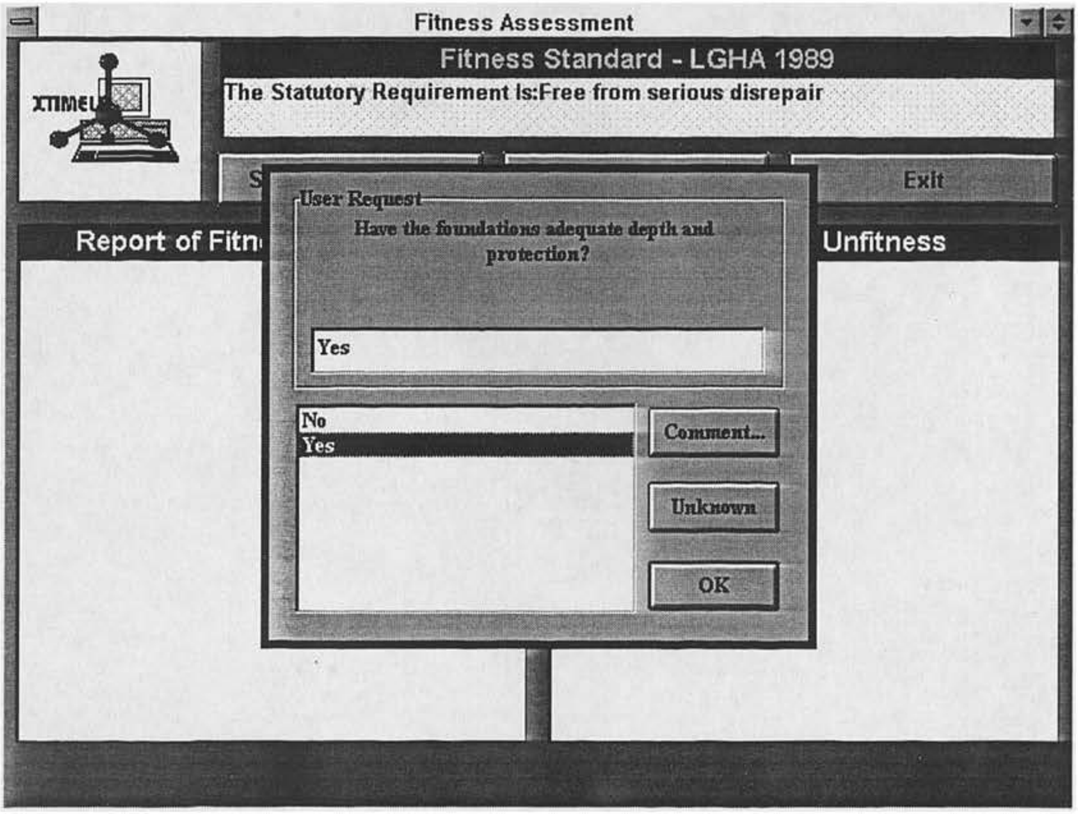
Screen 15: Input enquiry information-about the purpose of works



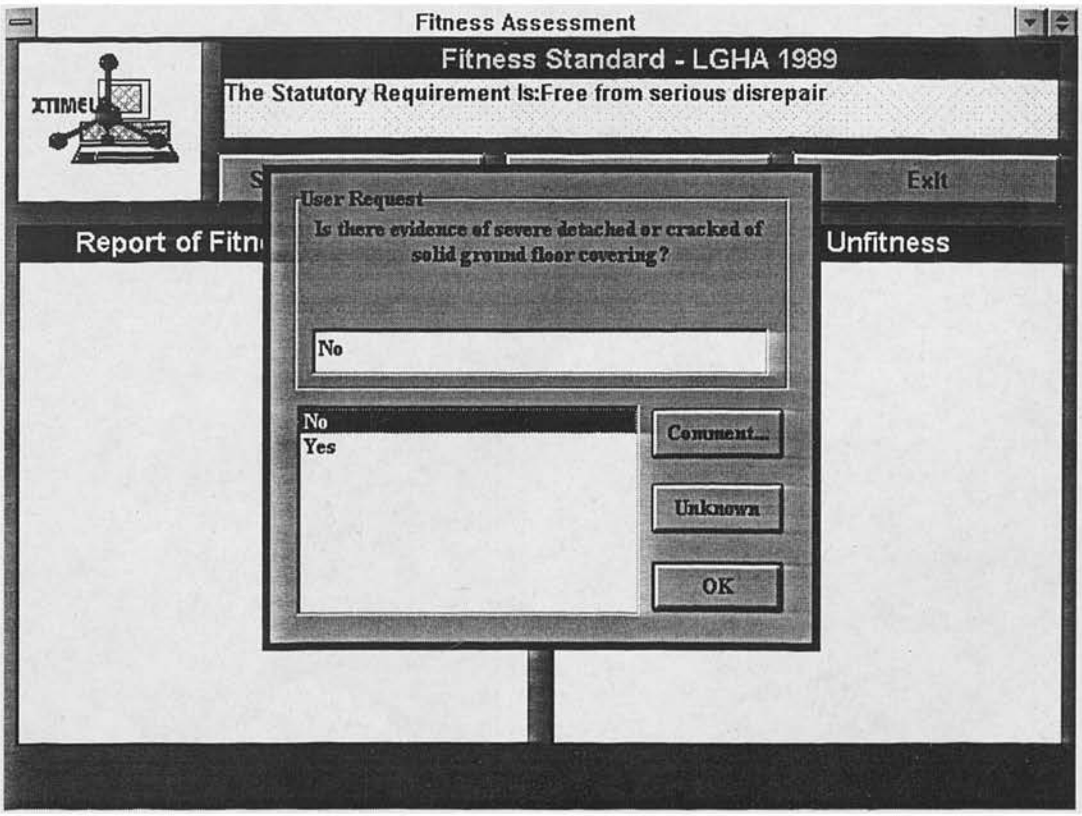
Screen 16: Report of the enquiry eligibility for an application for renovation grant

10.4- ASSESSMENT OF THE FITNESS OF THE PROPERTY

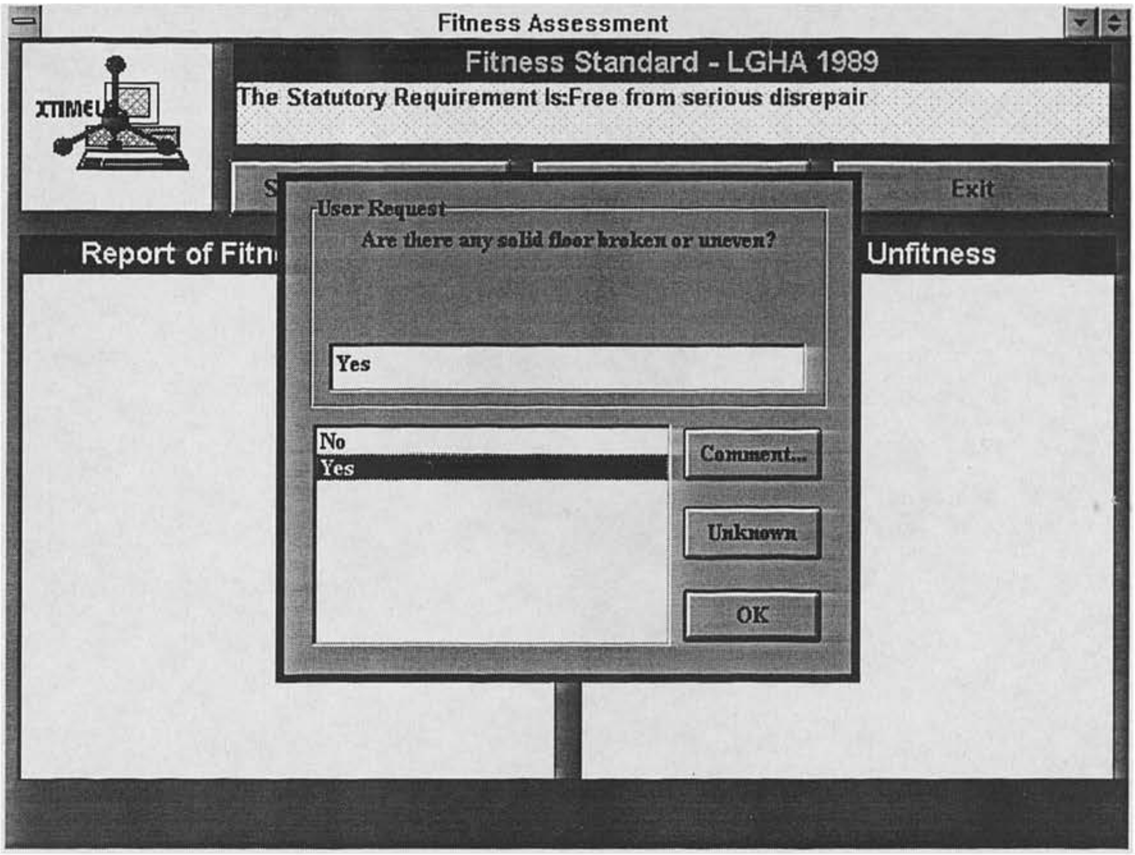
ASSESSING THE REPAIR REQUIREMENT



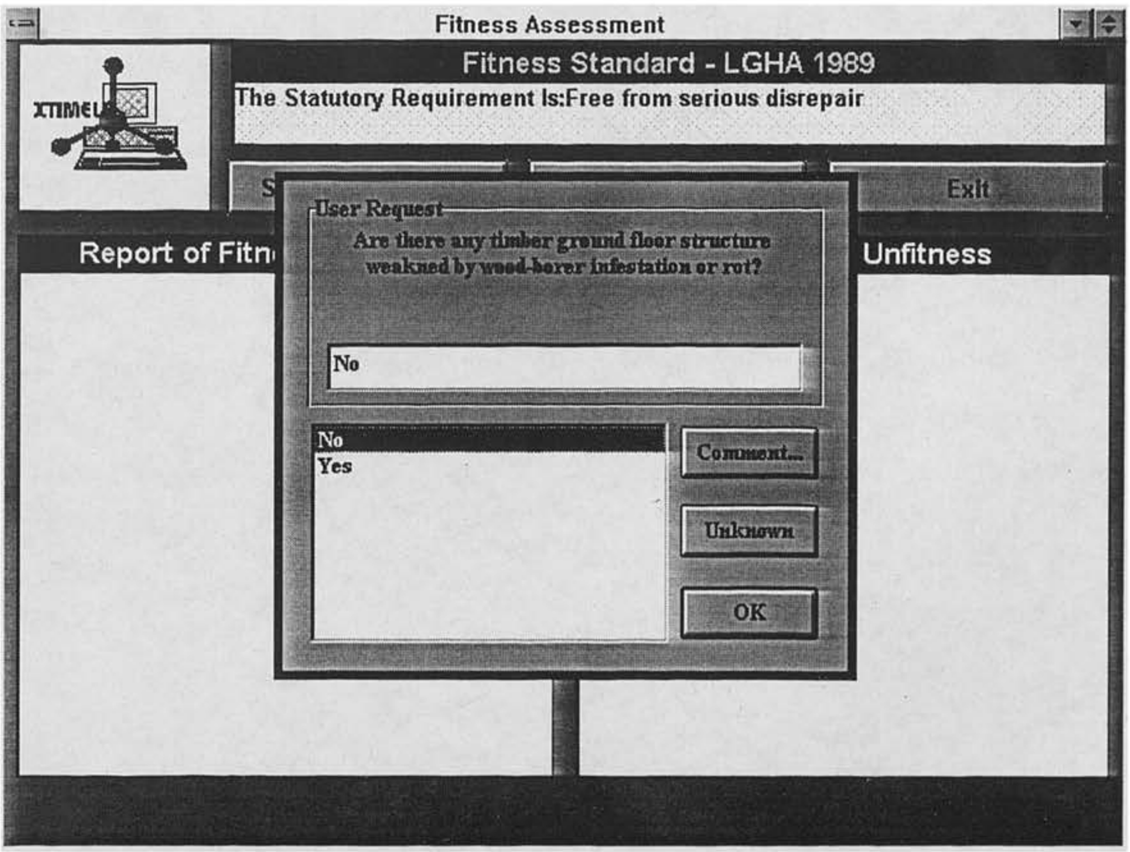
Screen 17: Input information about the condition of the building components



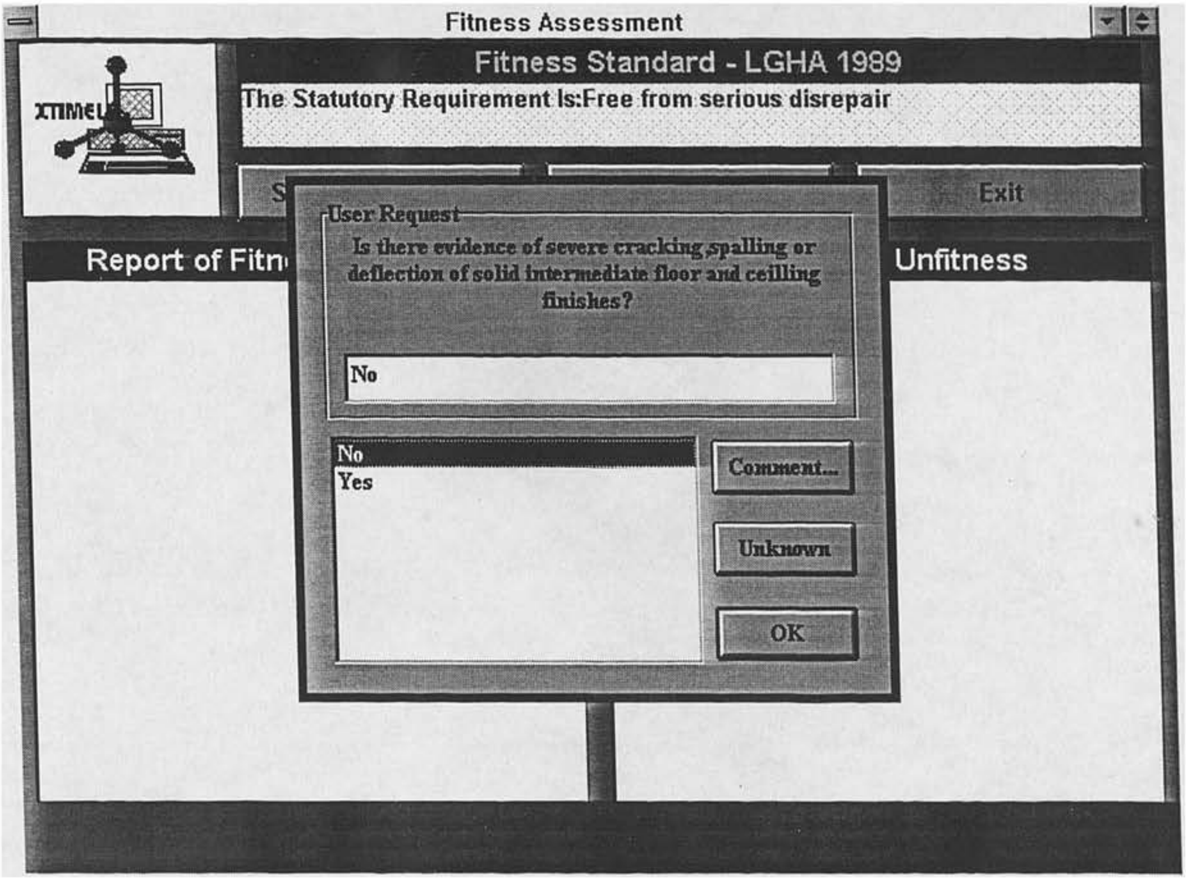
Screen 18: Input information about the condition of the building components



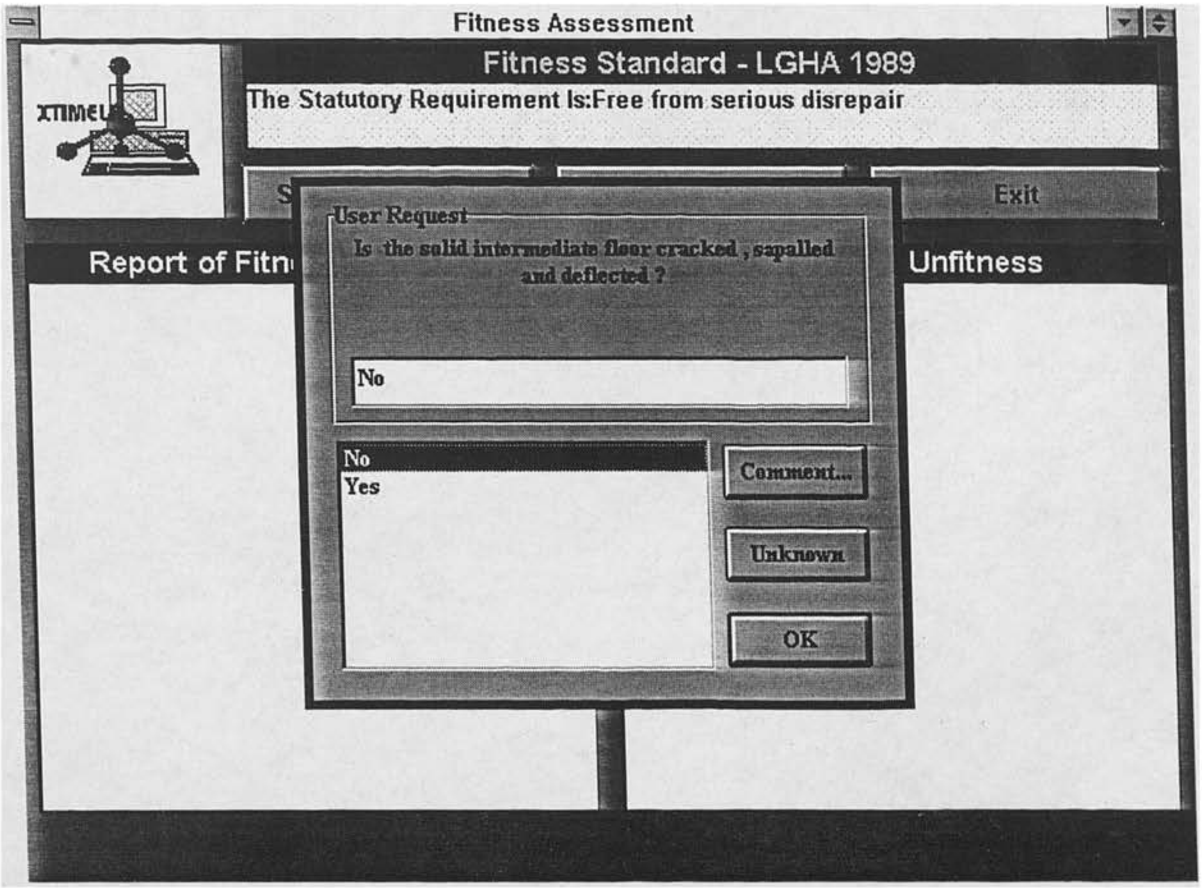
Screen 19: Input information about the condition of the building components



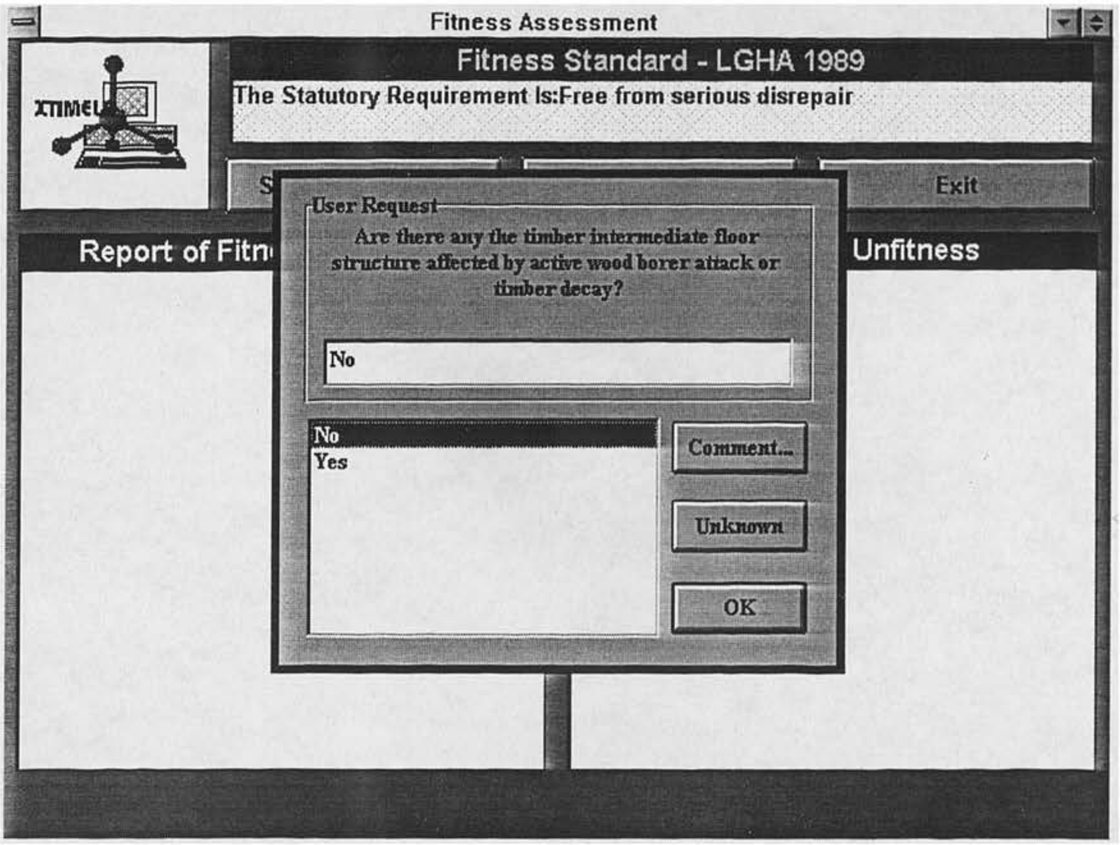
Screen 20: Input information about the condition of the building components



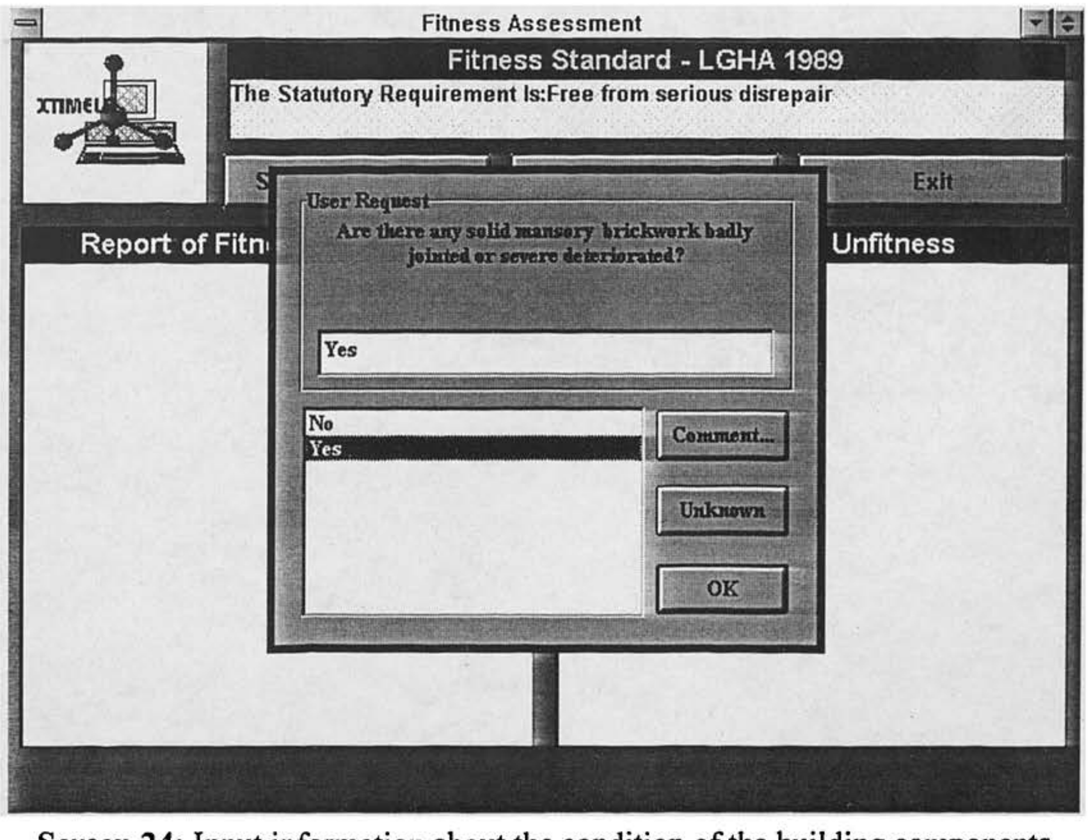
Screen 21: Input information about the condition of the building components



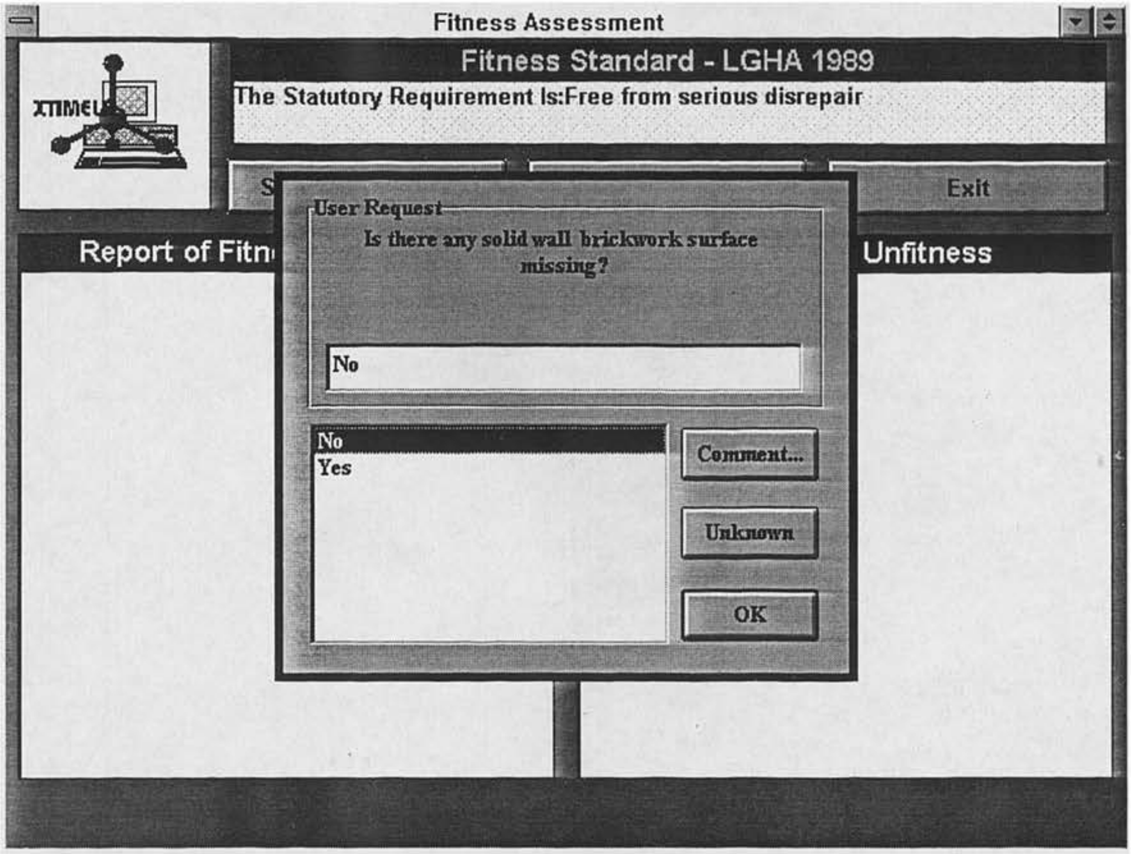
Screen 22: Input information about the condition of the building components



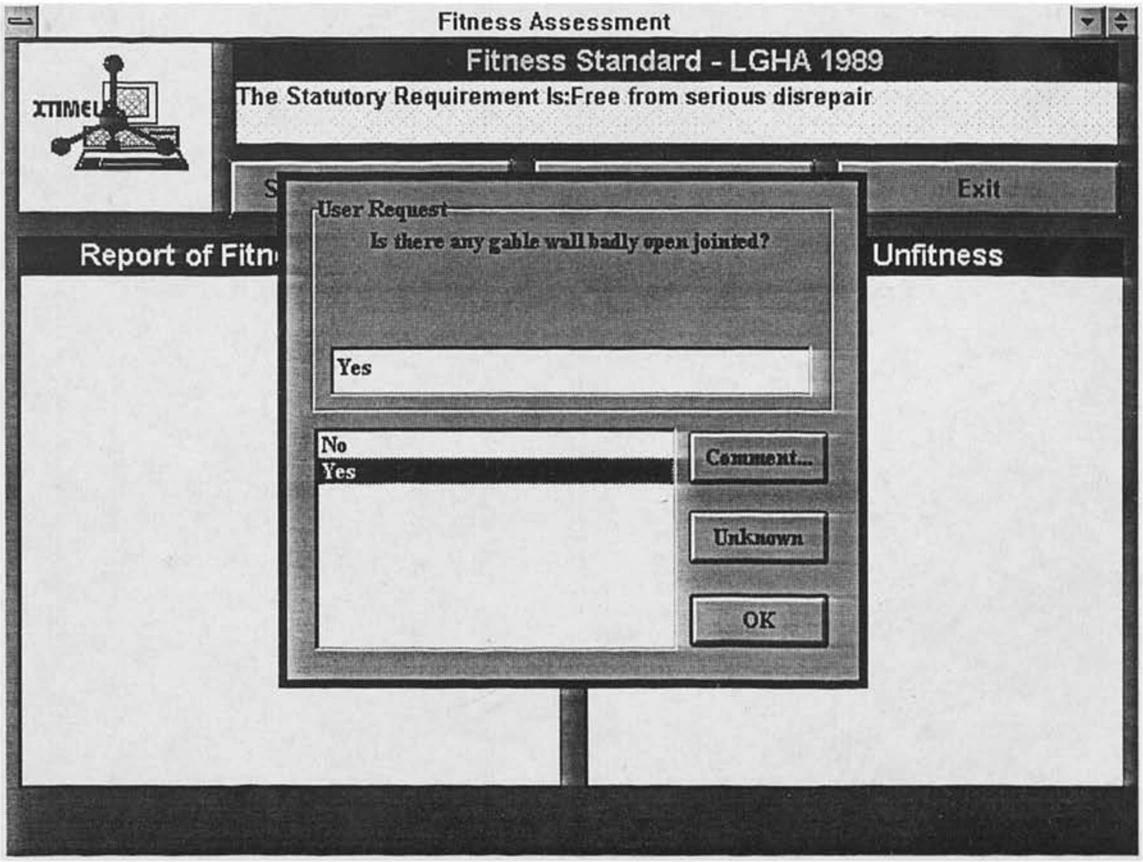
Screen 23: Input information about the condition of the building components



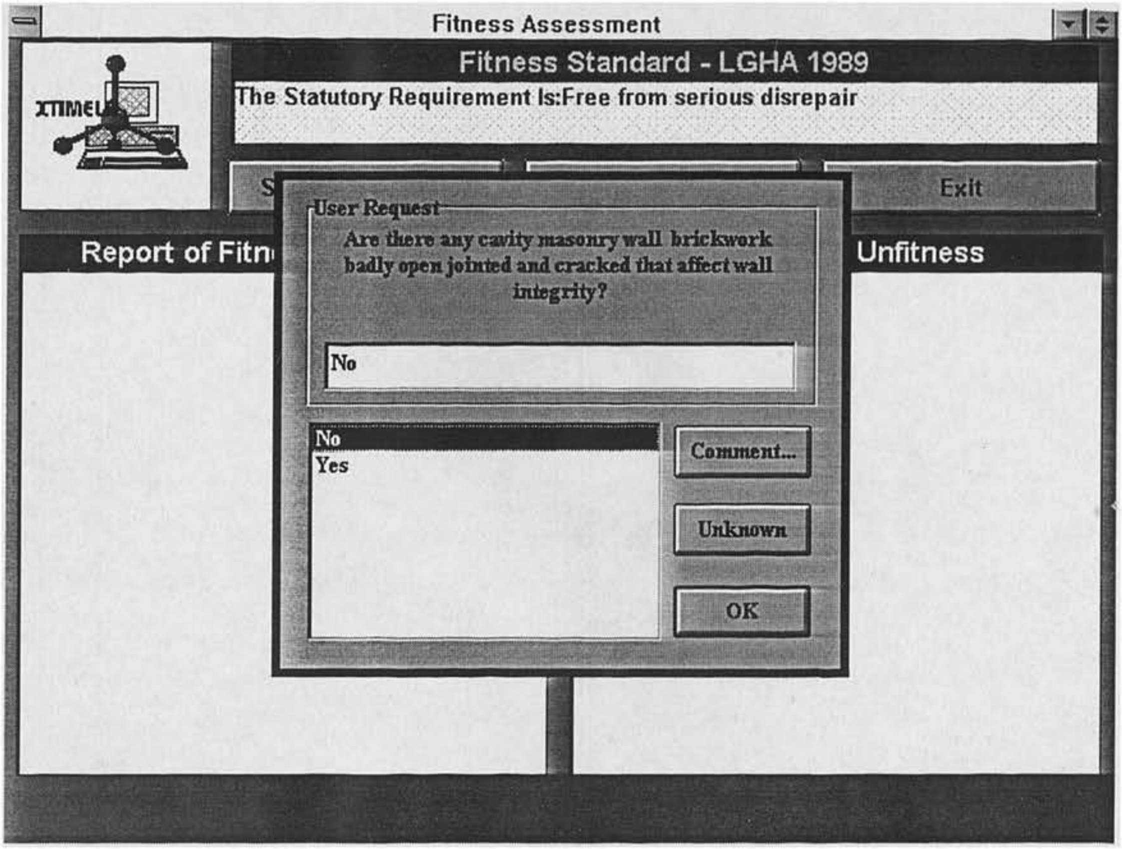
Screen 24: Input information about the condition of the building components



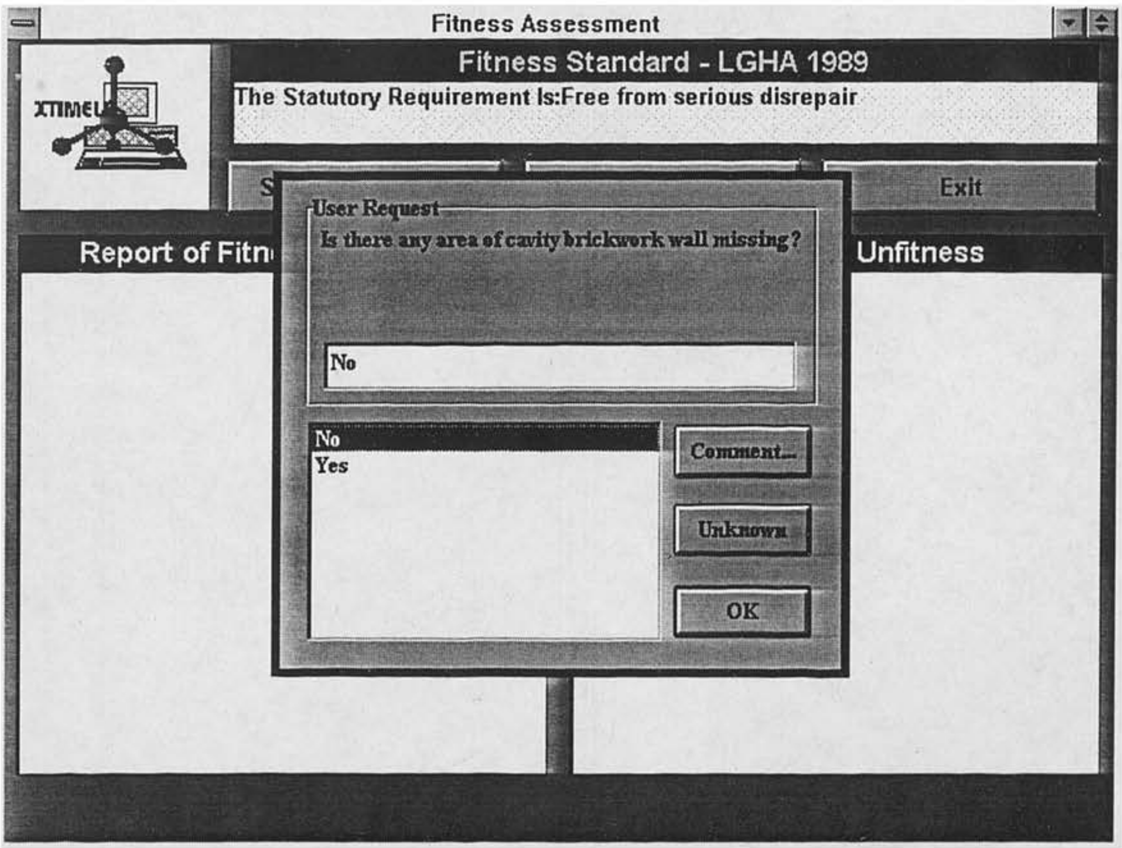
Screen 25: Input information about the condition of the building components



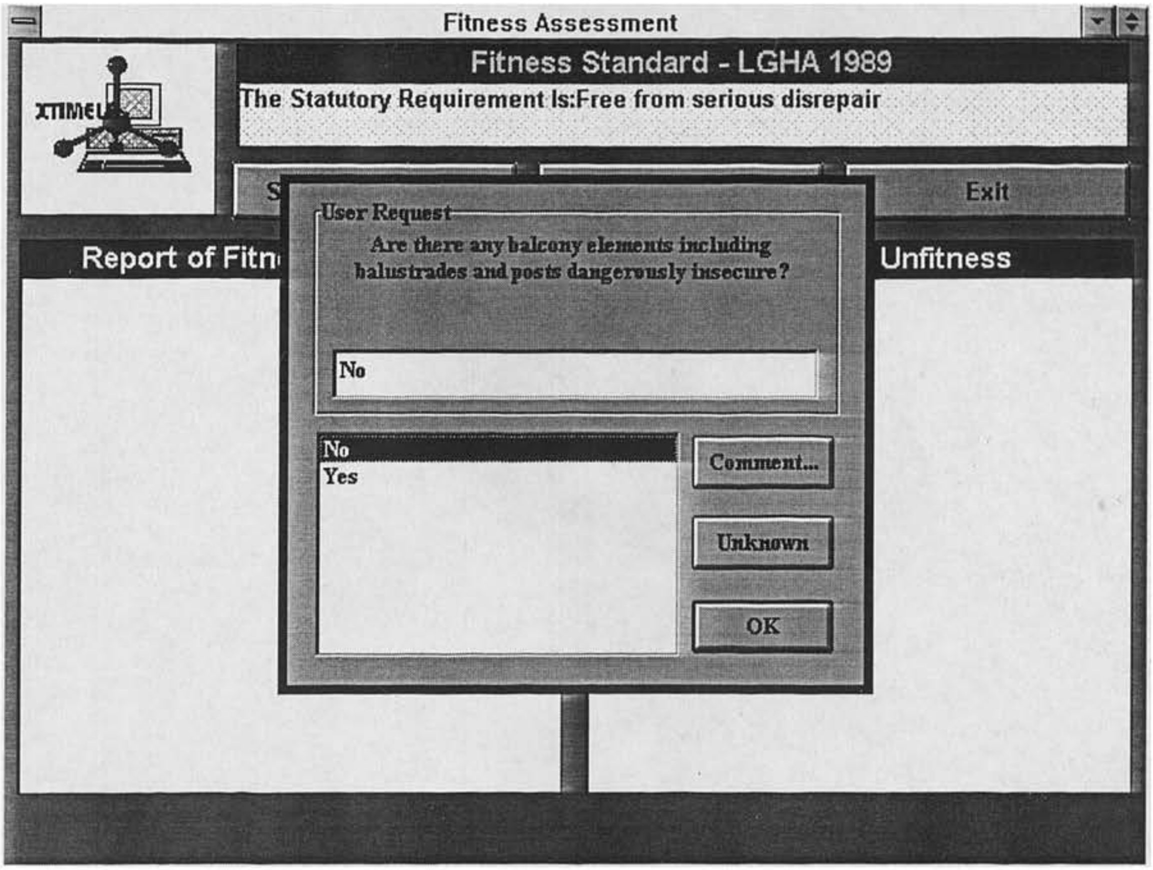
Screen 26: Input information about the condition of the building components



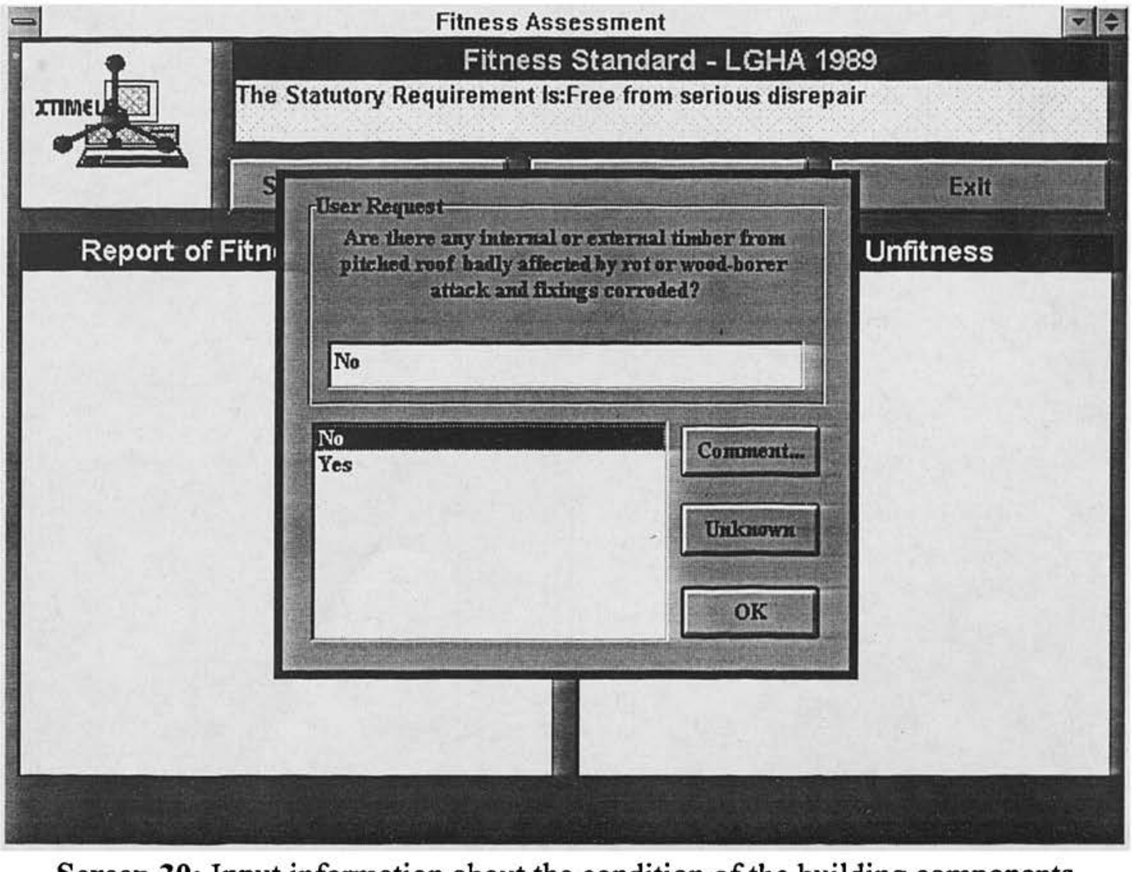
Screen 27: Input information about the condition of the building components



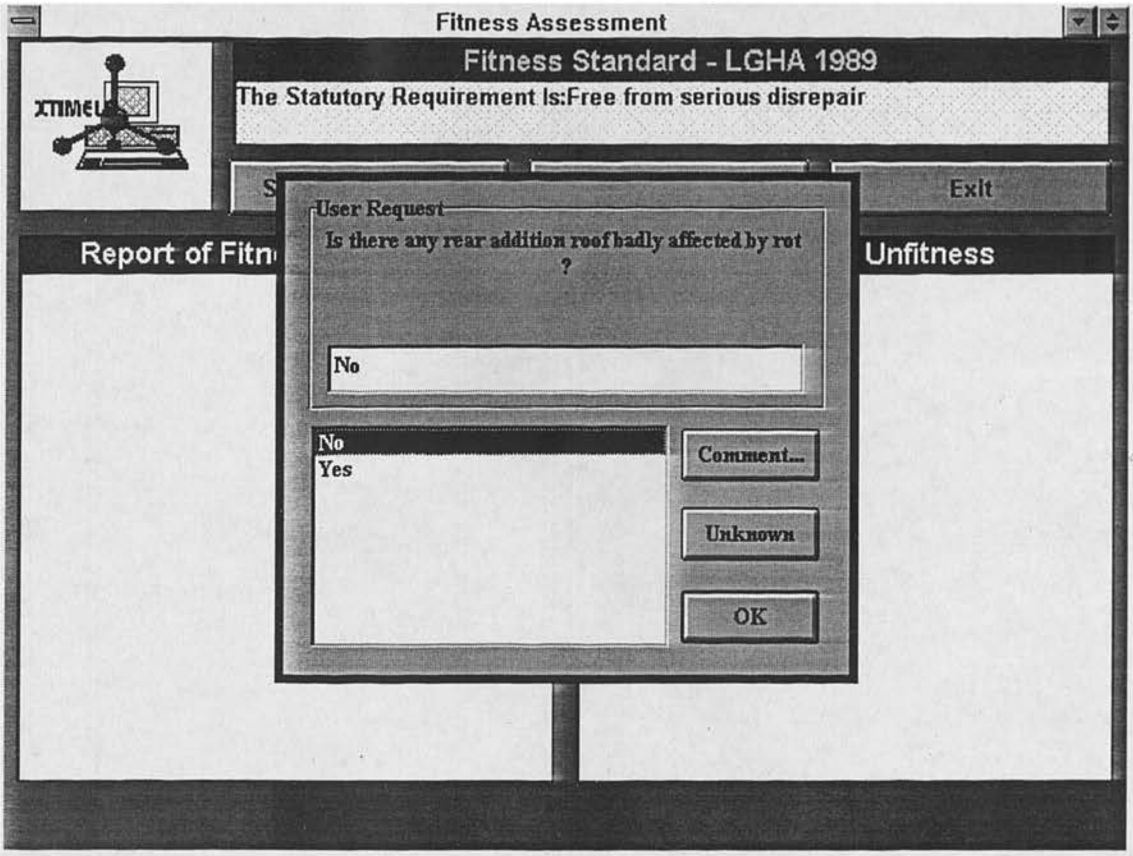
Screen 28: Input information about the condition of the building components



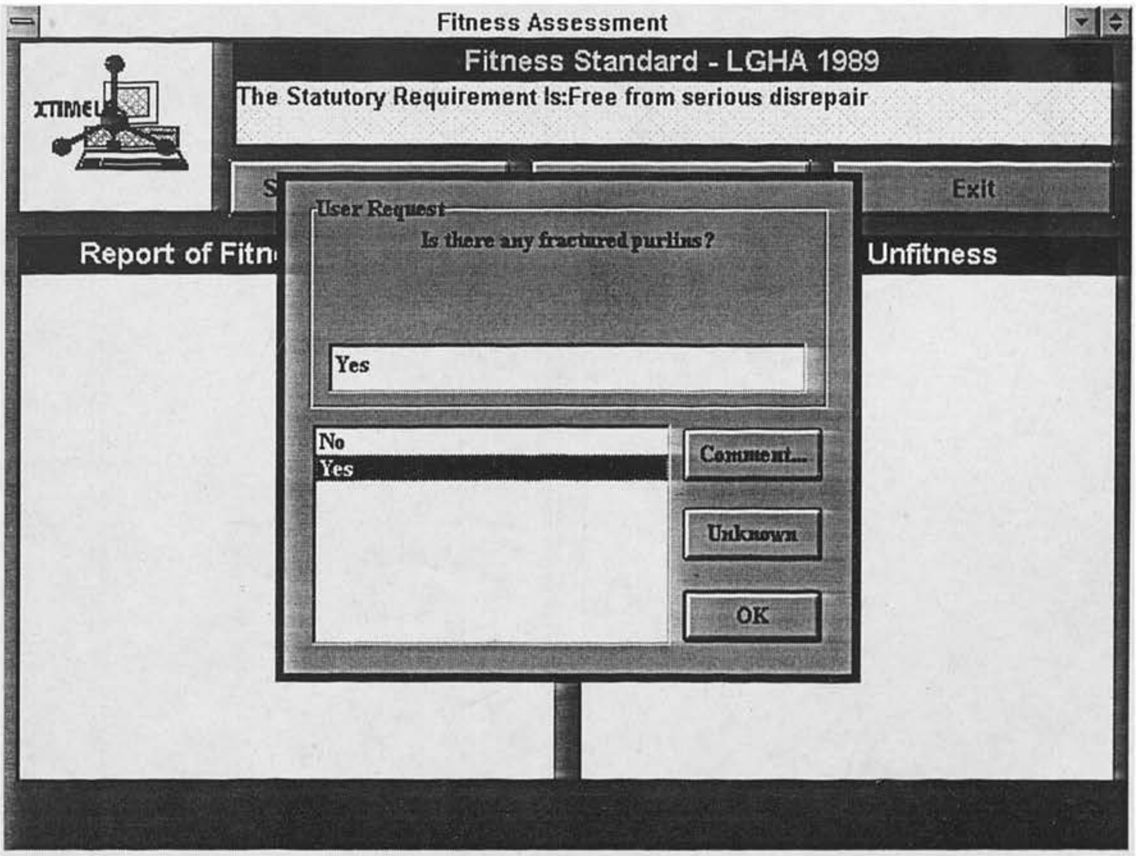
Screen 29: Input information about the condition of the building components



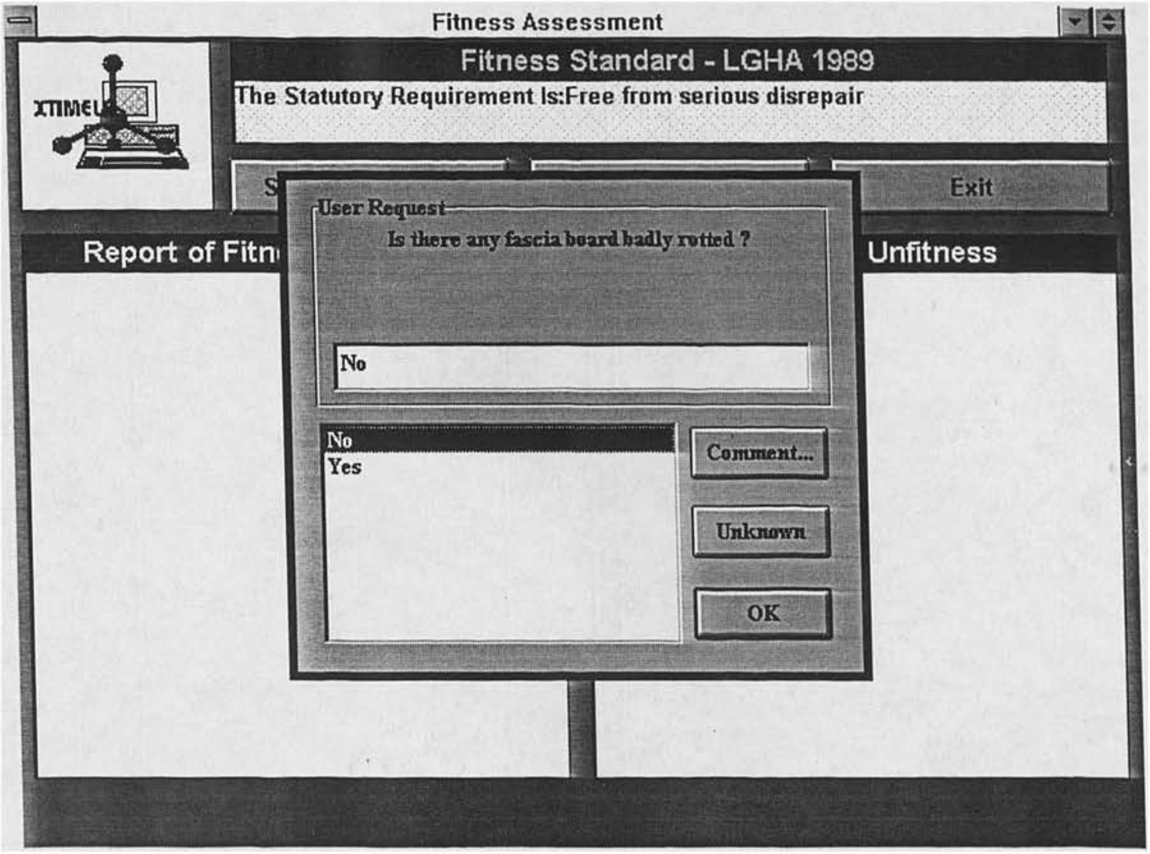
Screen 30: Input information about the condition of the building components



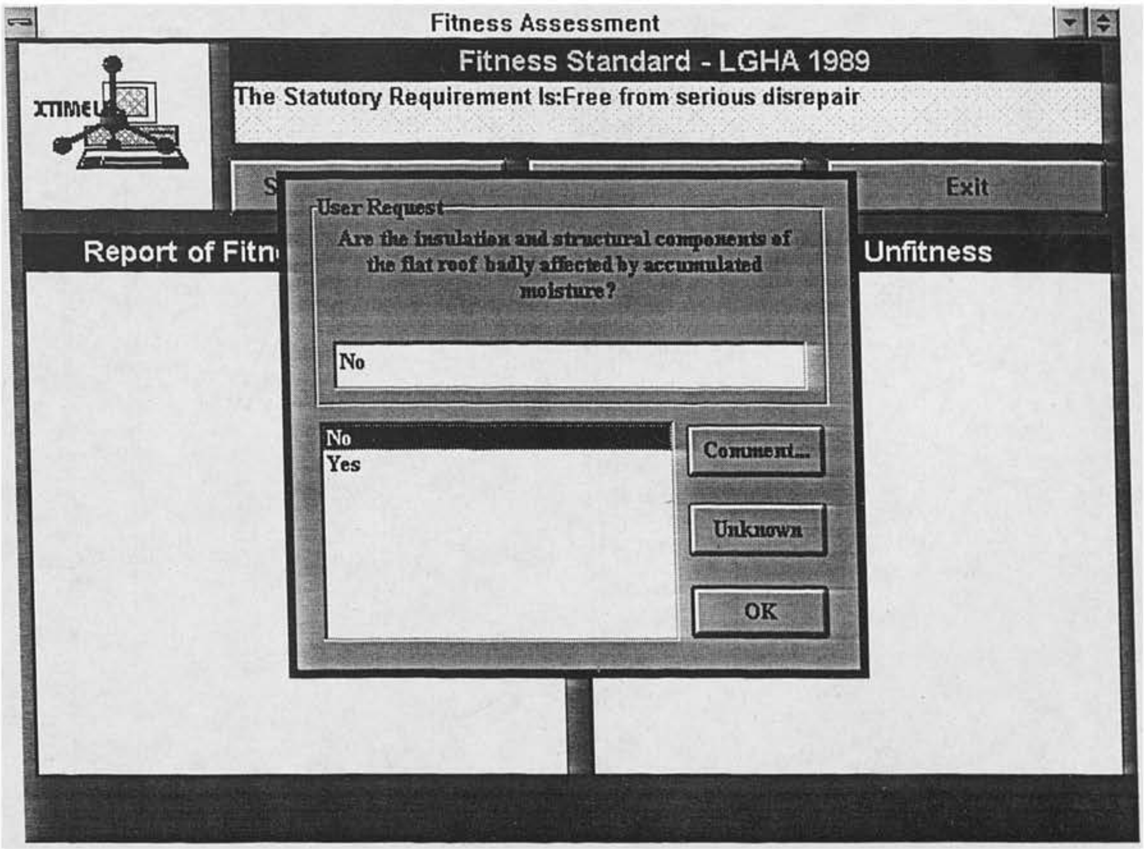
Screen 31: Input information about the condition of the building components



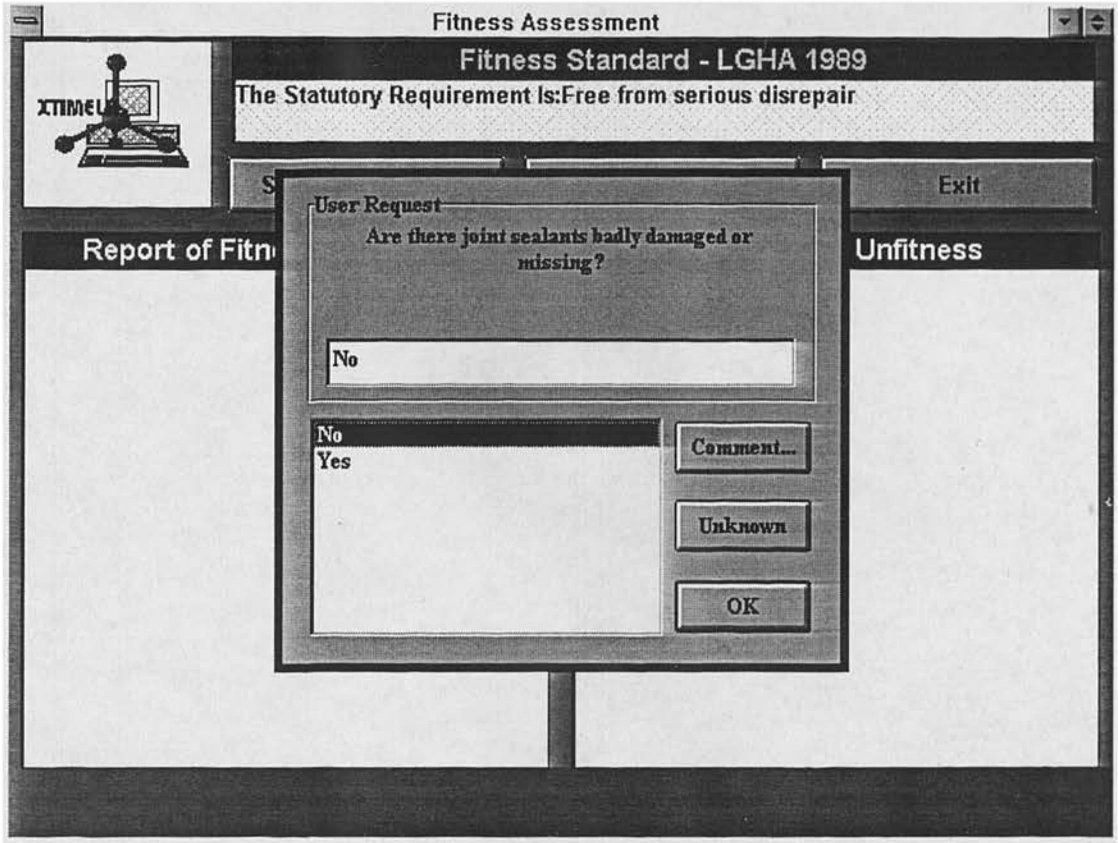
Screen 32: Input information about the condition of the building components



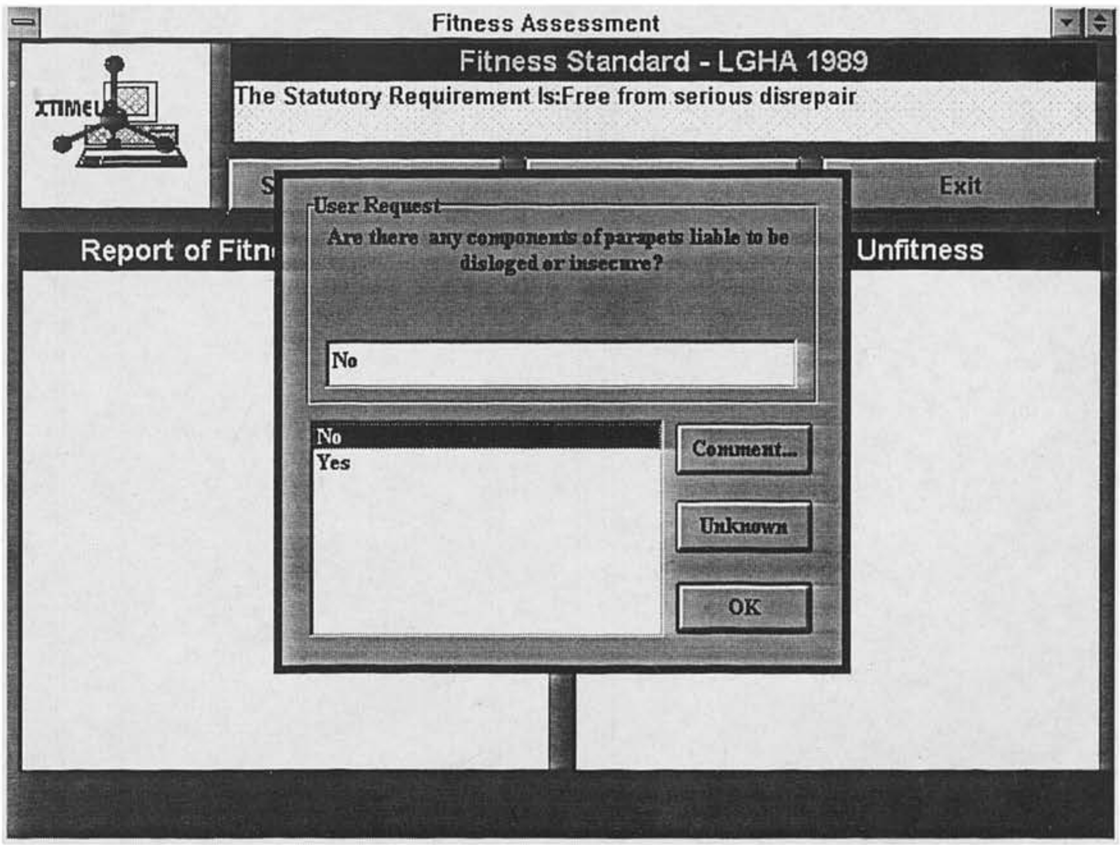
Screen 33: Input information about the condition of the building components



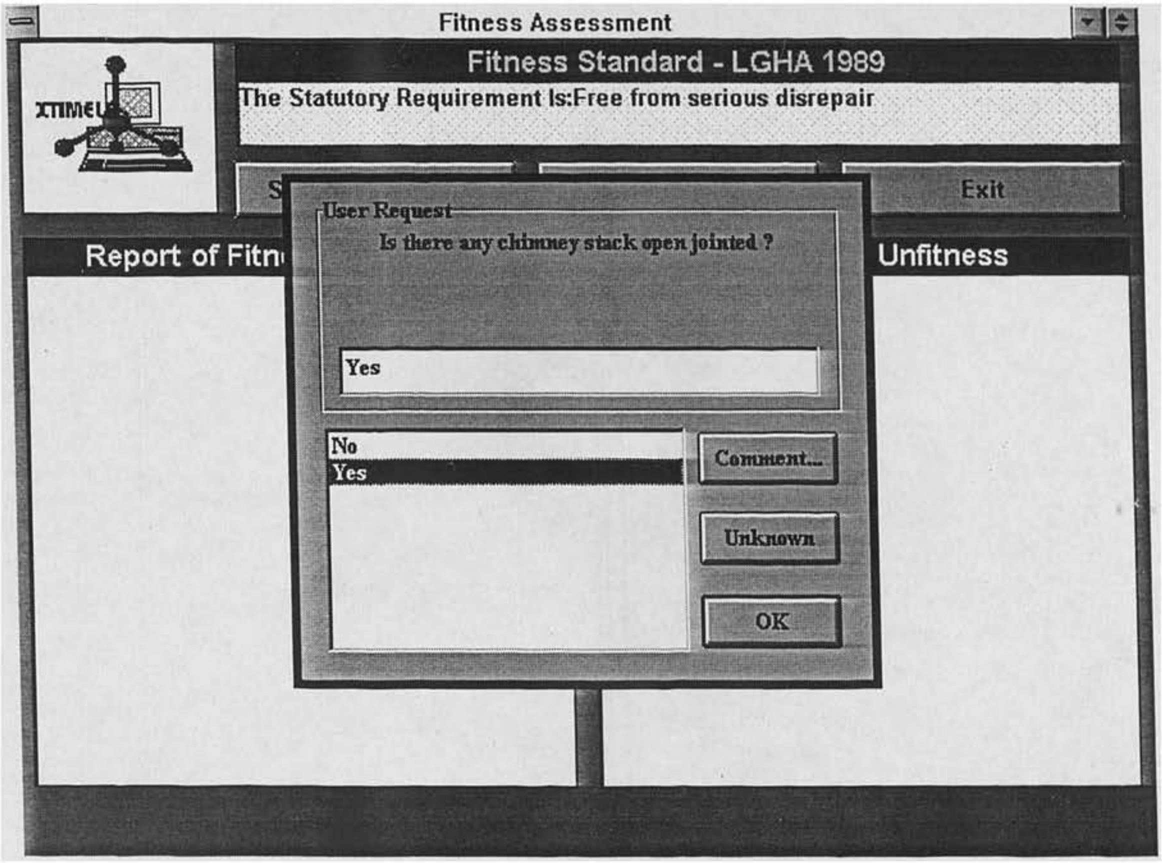
Screen 34: Input information about the condition of the building components



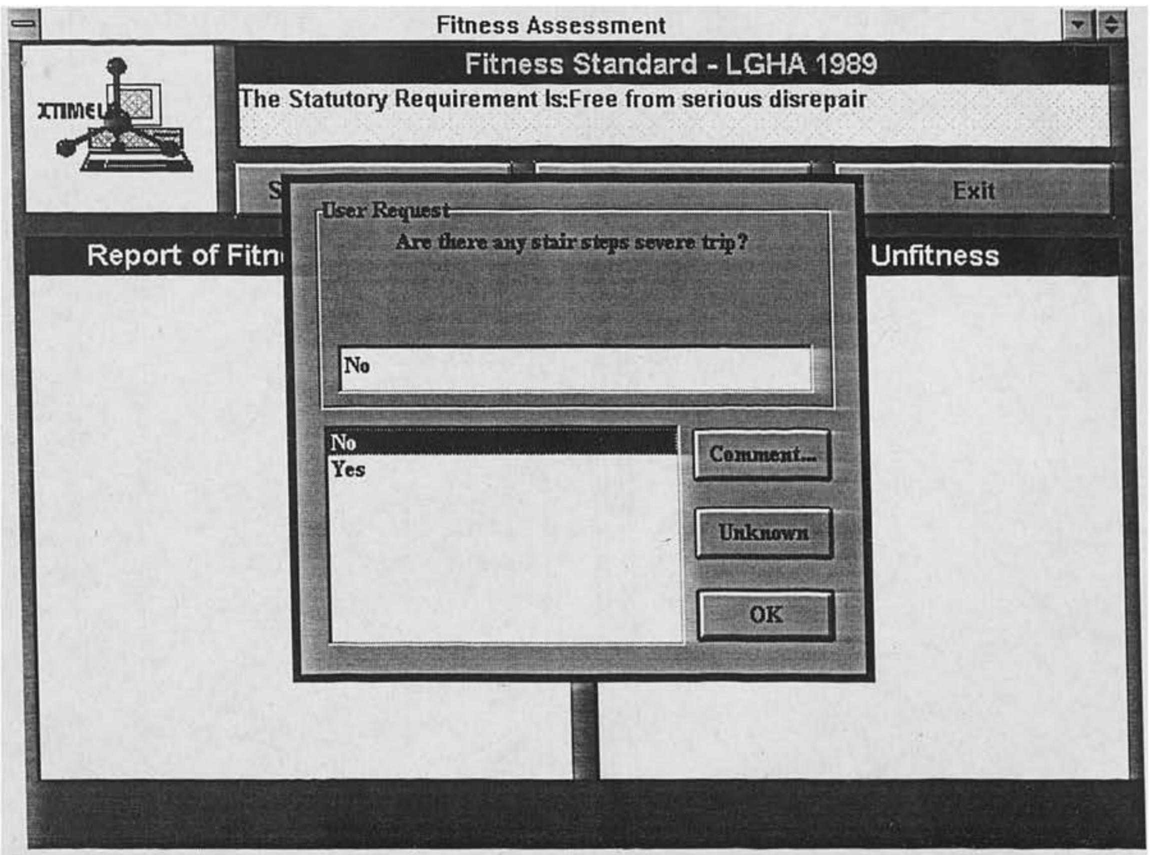
Screen 35: Input information about the condition of the building components



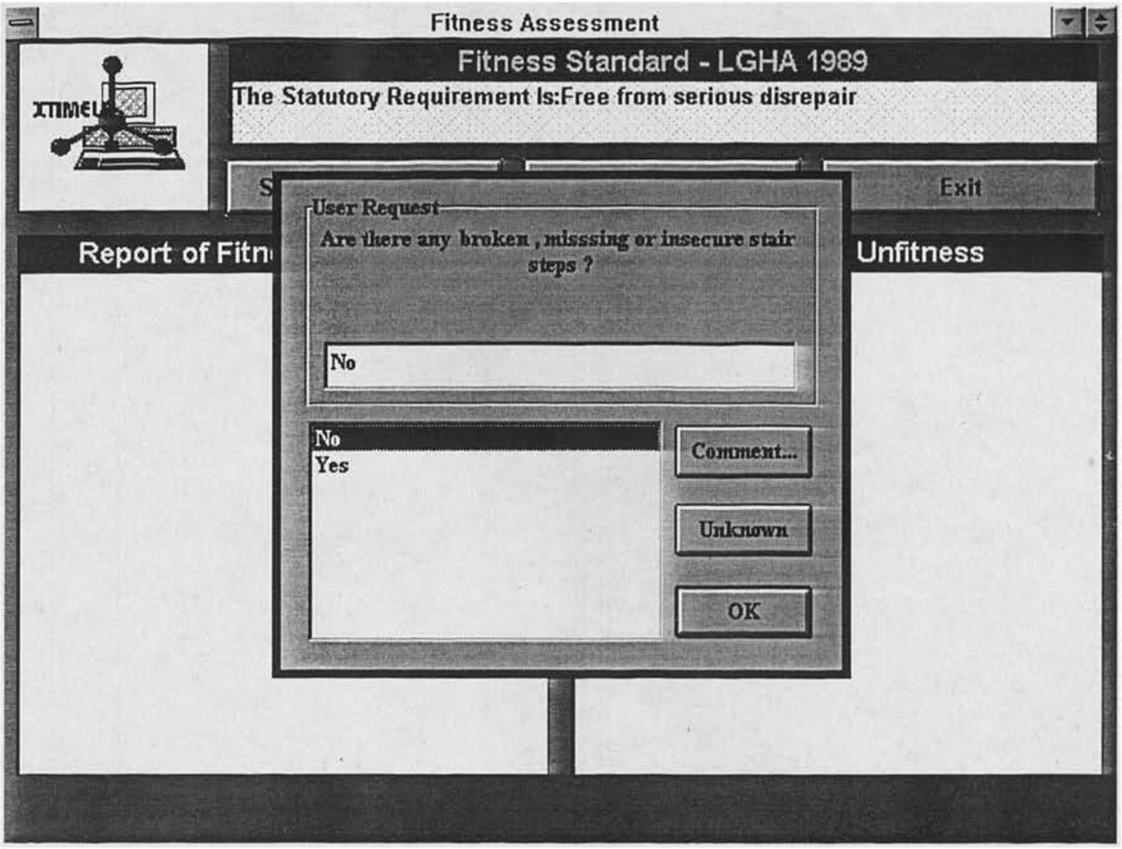
Screen 36: Input information about the condition of the building components



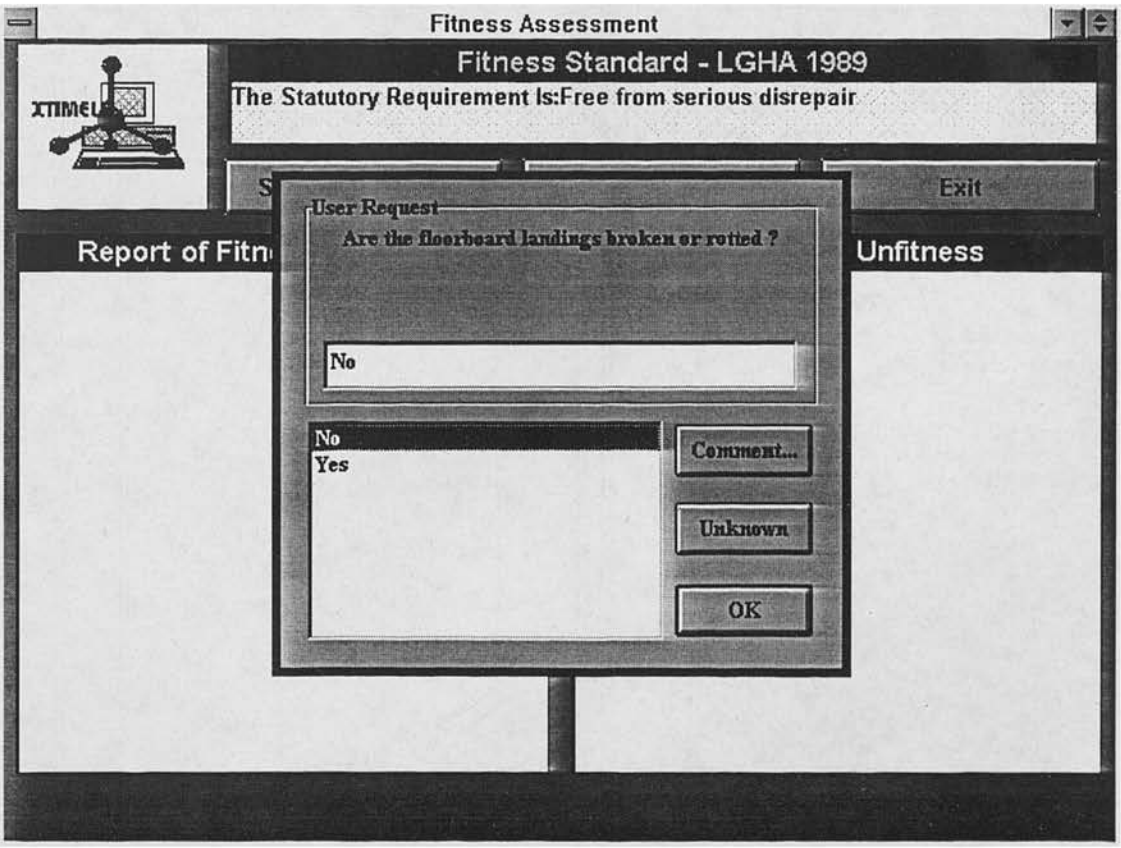
Screen 37: Input information about the condition of the building components



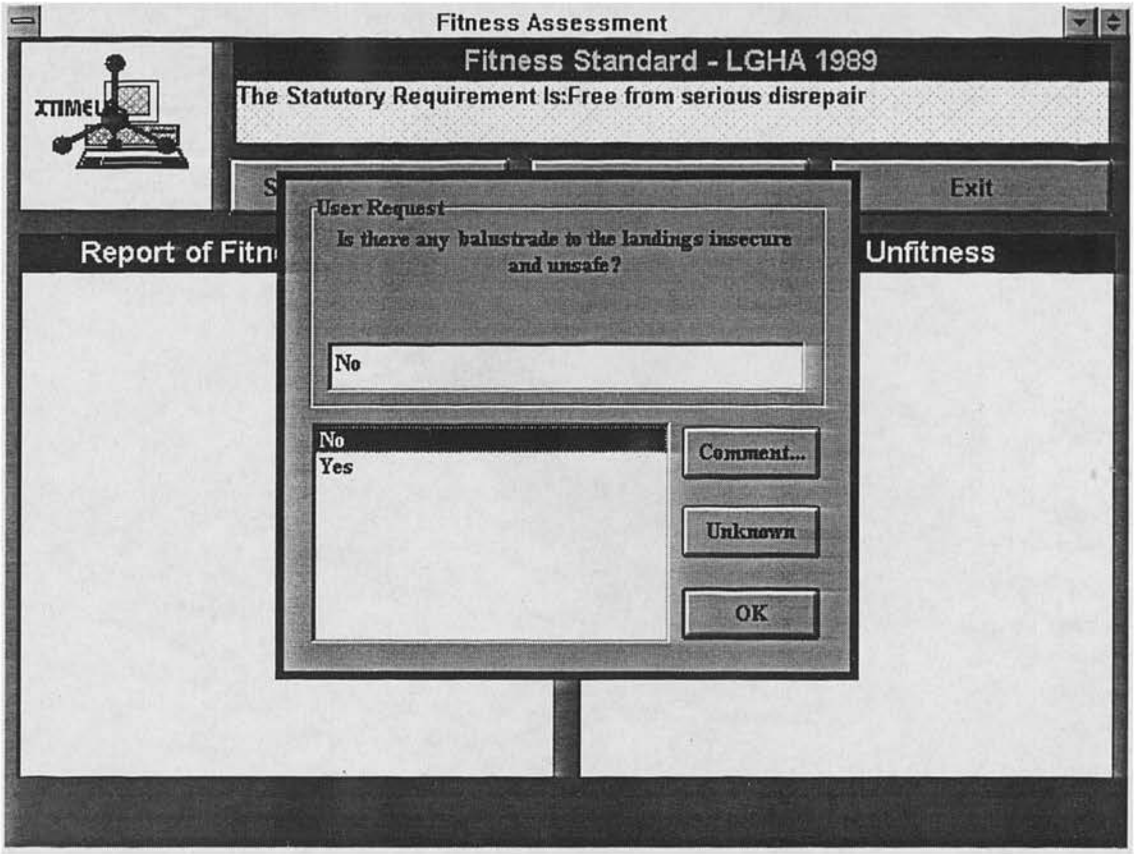
Screen 38: Input information about the condition of the building components



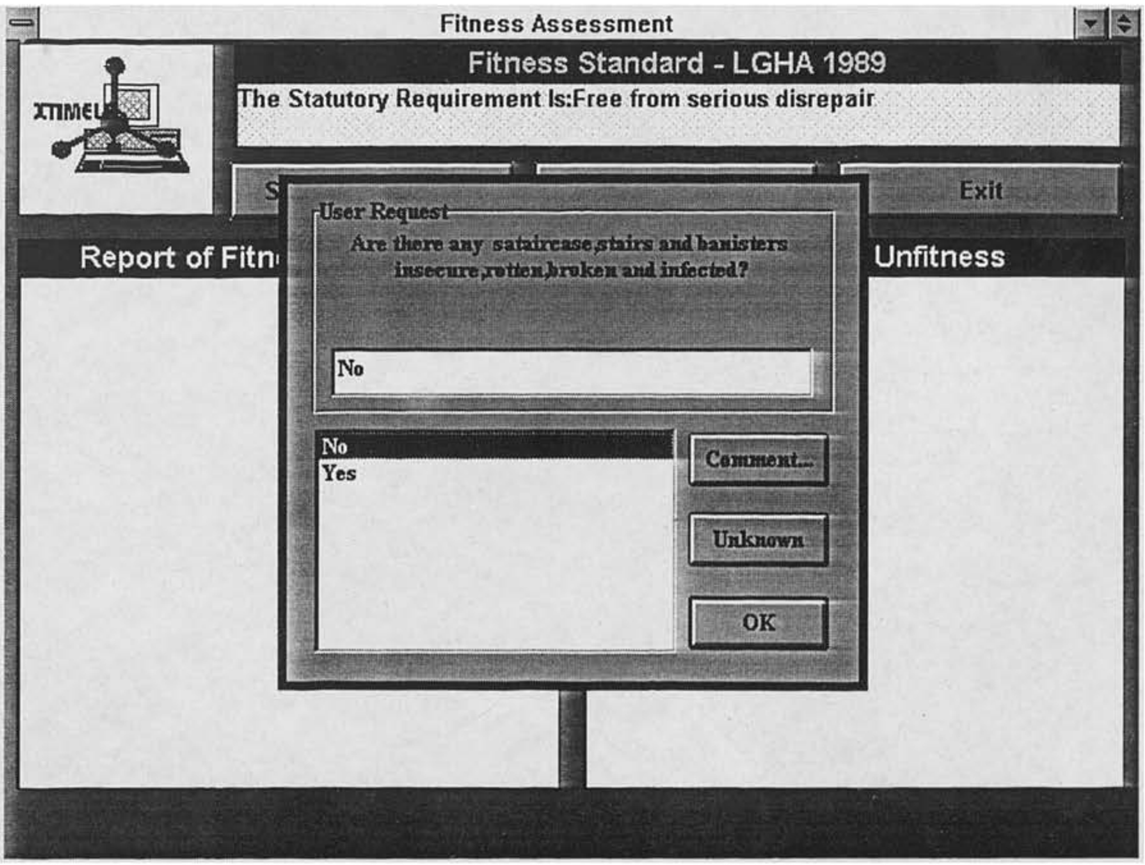
Screen 39: Input information about the condition of the building components



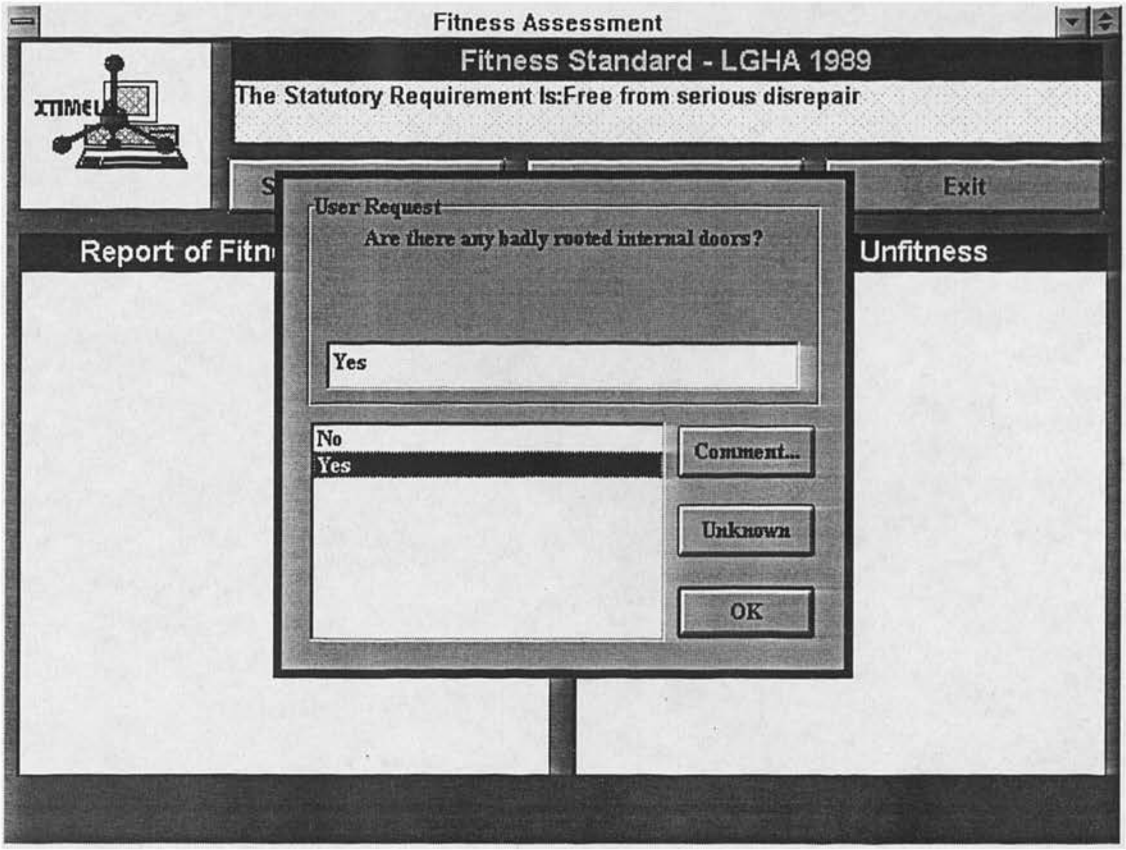
Screen 40: Input information about the condition of the building components



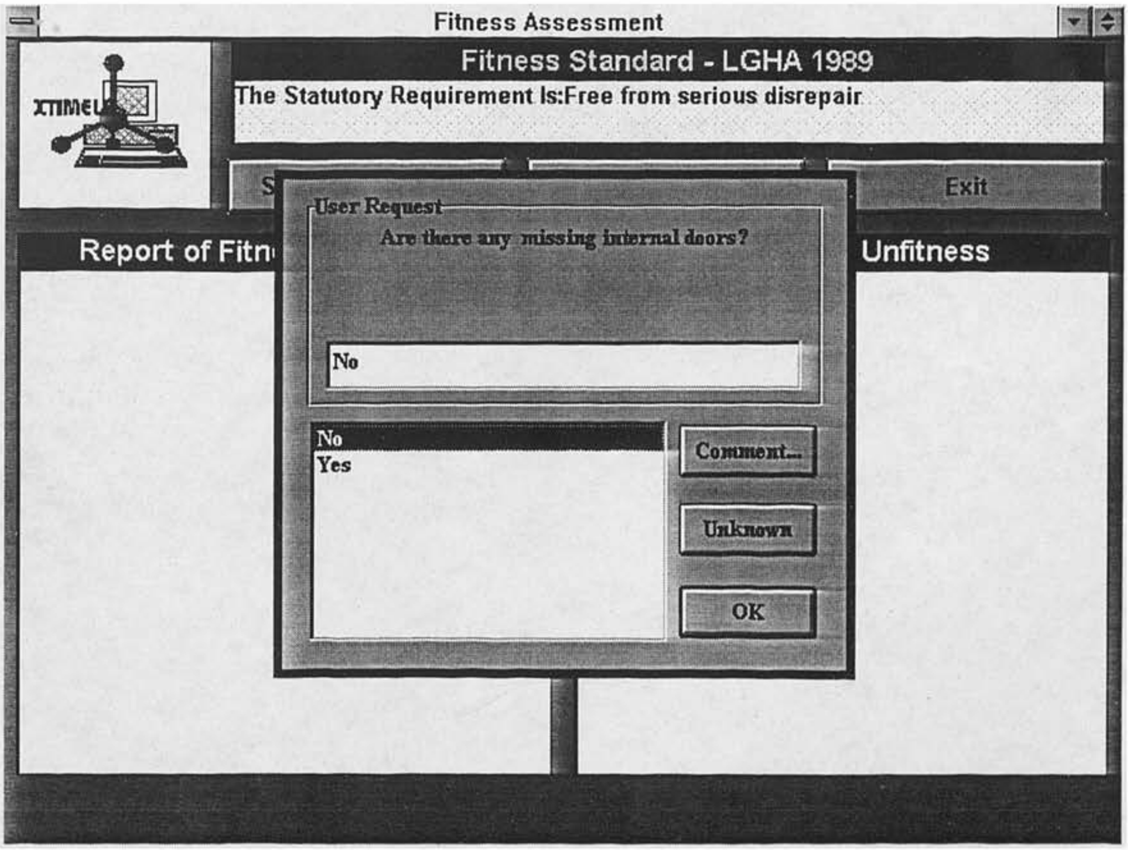
Screen 41: Input information about the condition of the building components



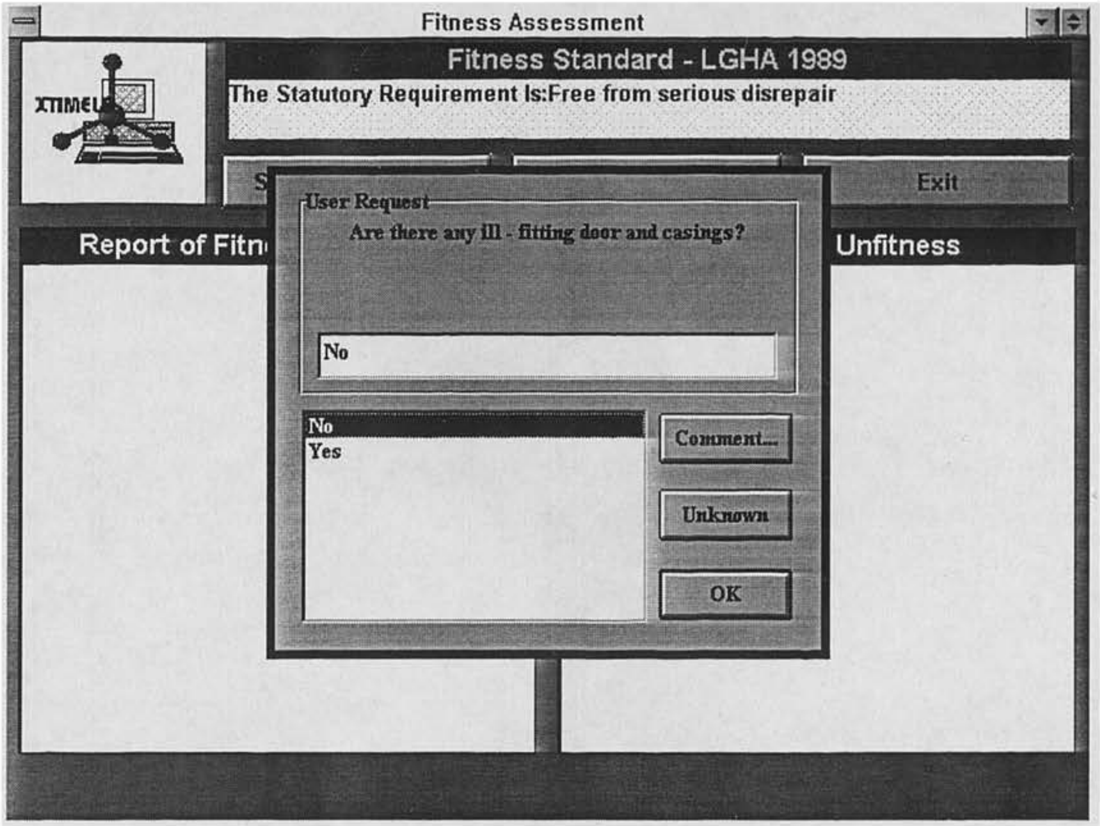
Screen 42: Input information about the condition of the building components



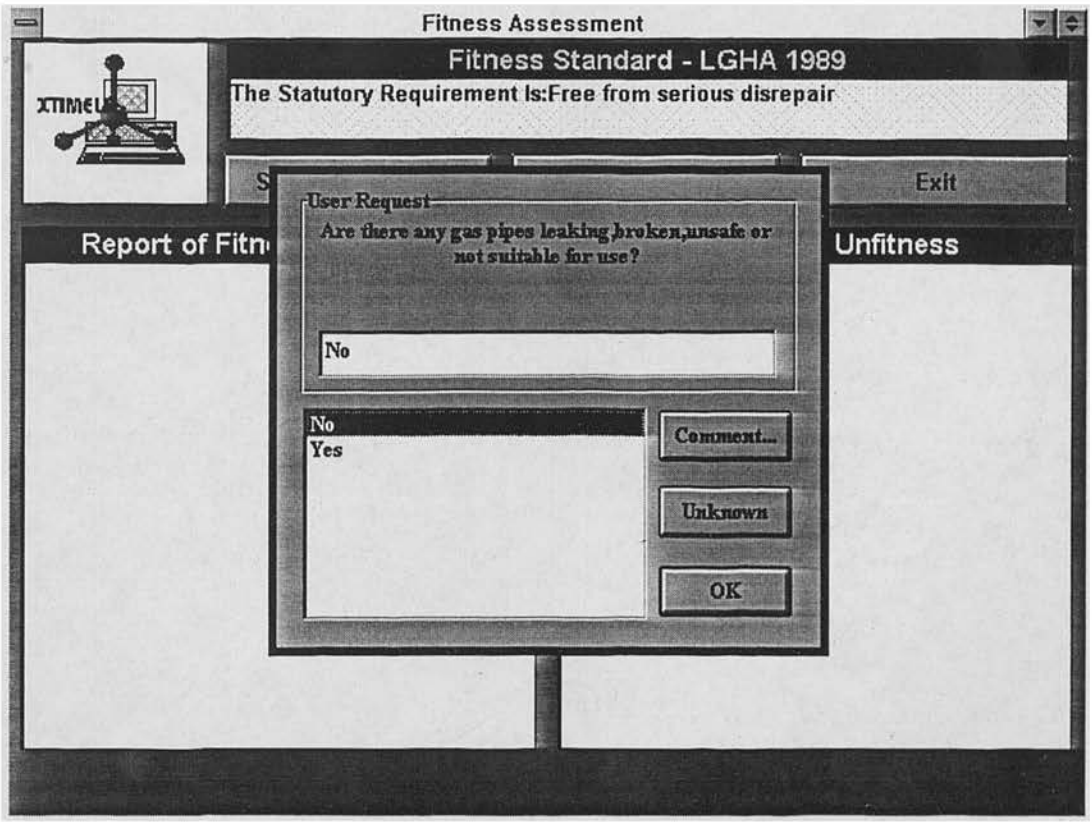
Screen 43: Input information about the condition of the building components



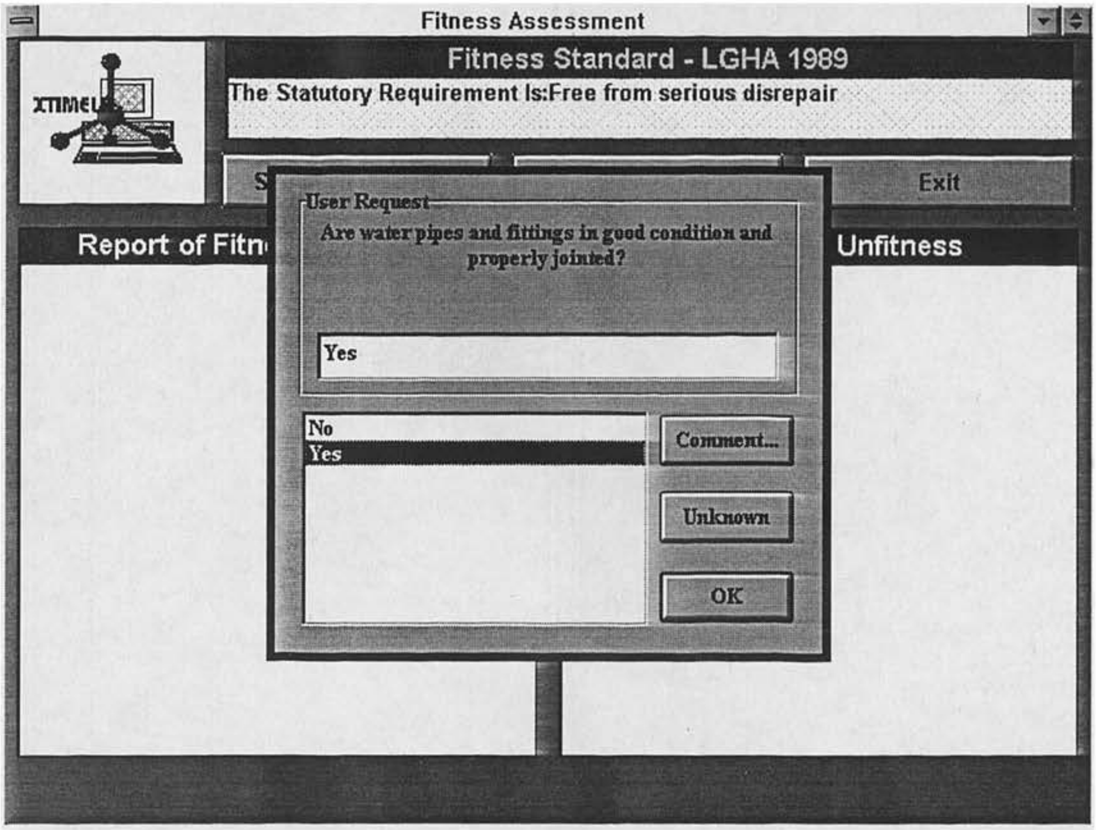
Screen 44: Input information about the condition of the building components



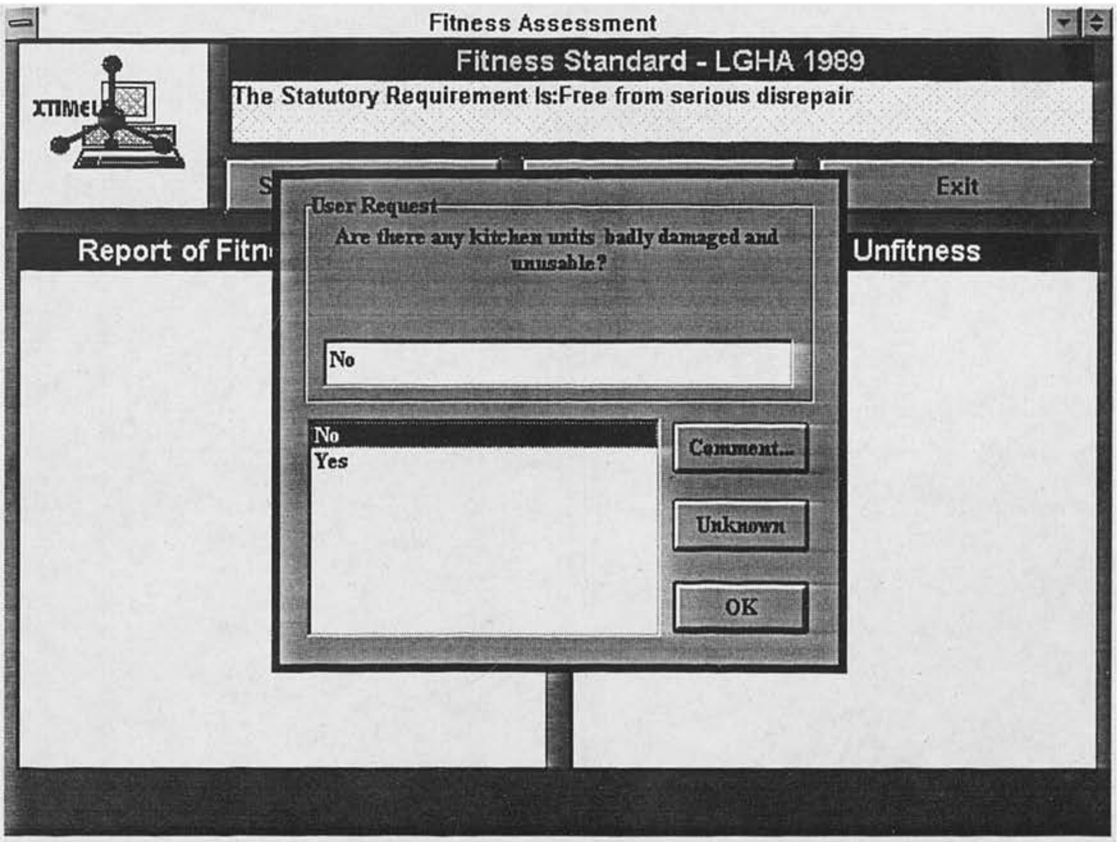
Screen 45: Input information about the condition of the building components



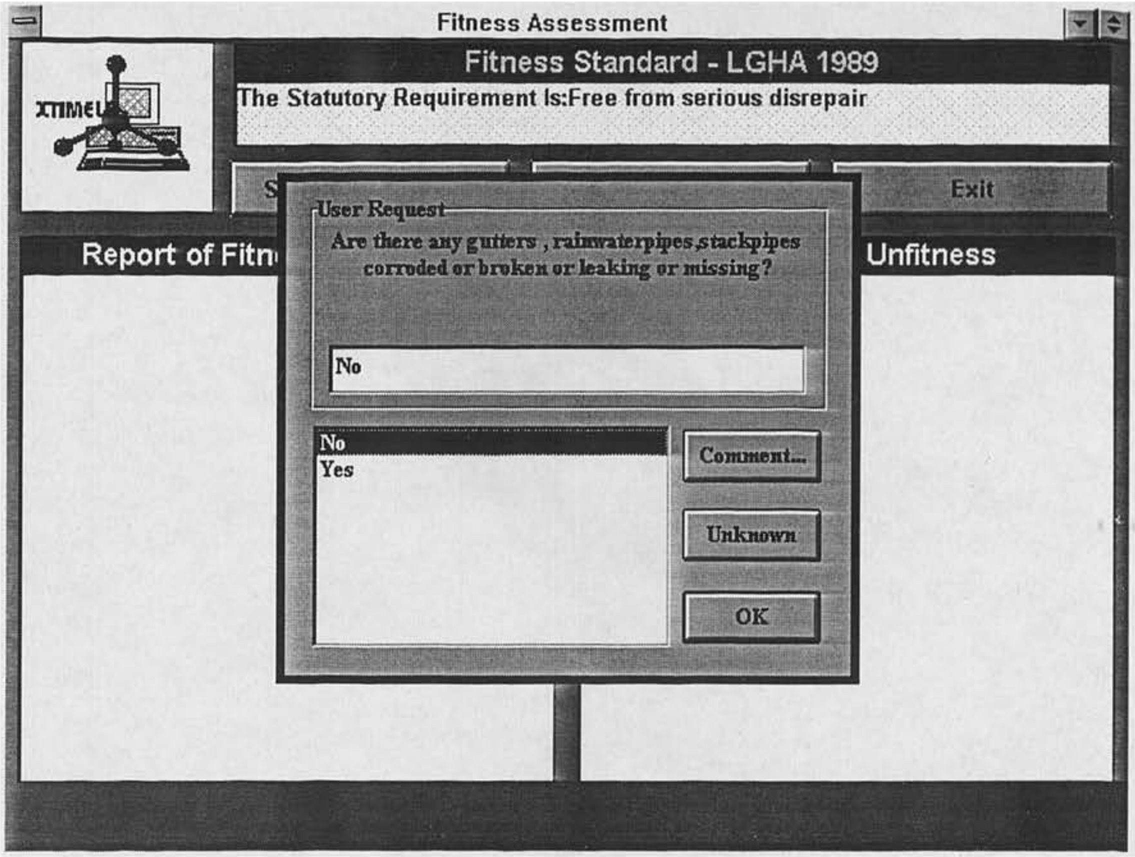
Screen 46: Input information about the condition of the building components



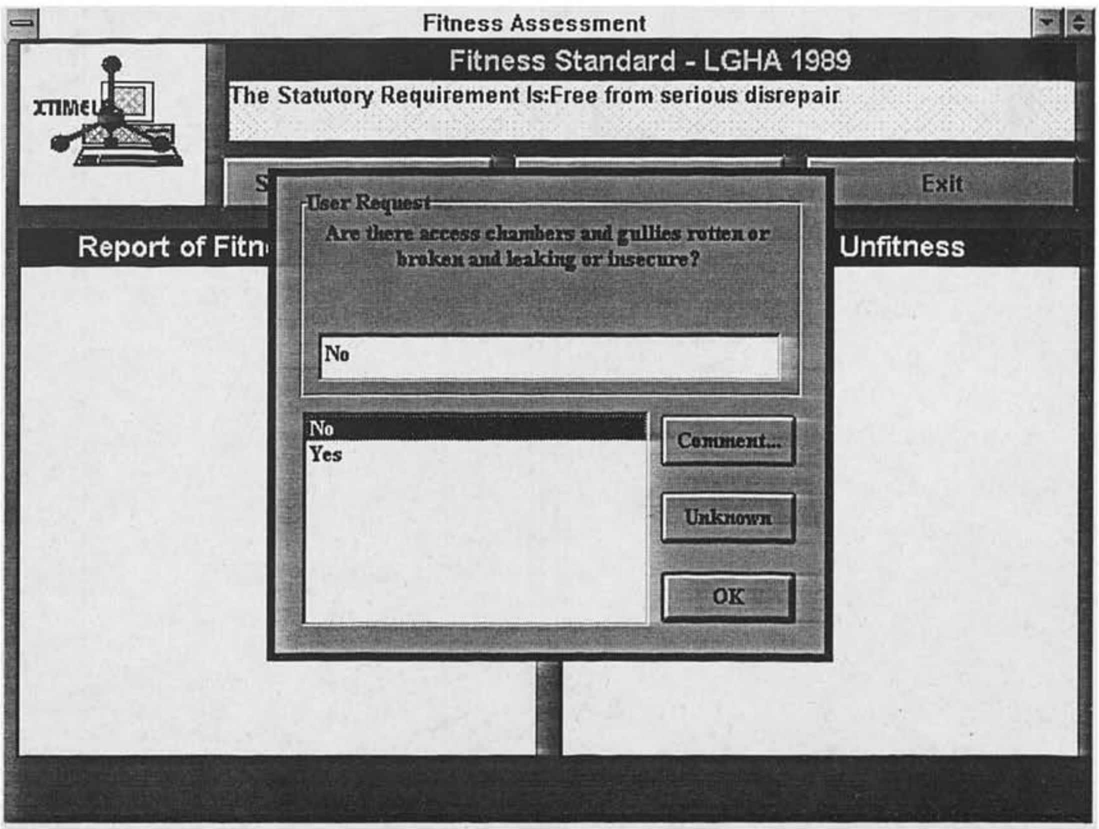
Screen 47: Input information about the condition of the building components



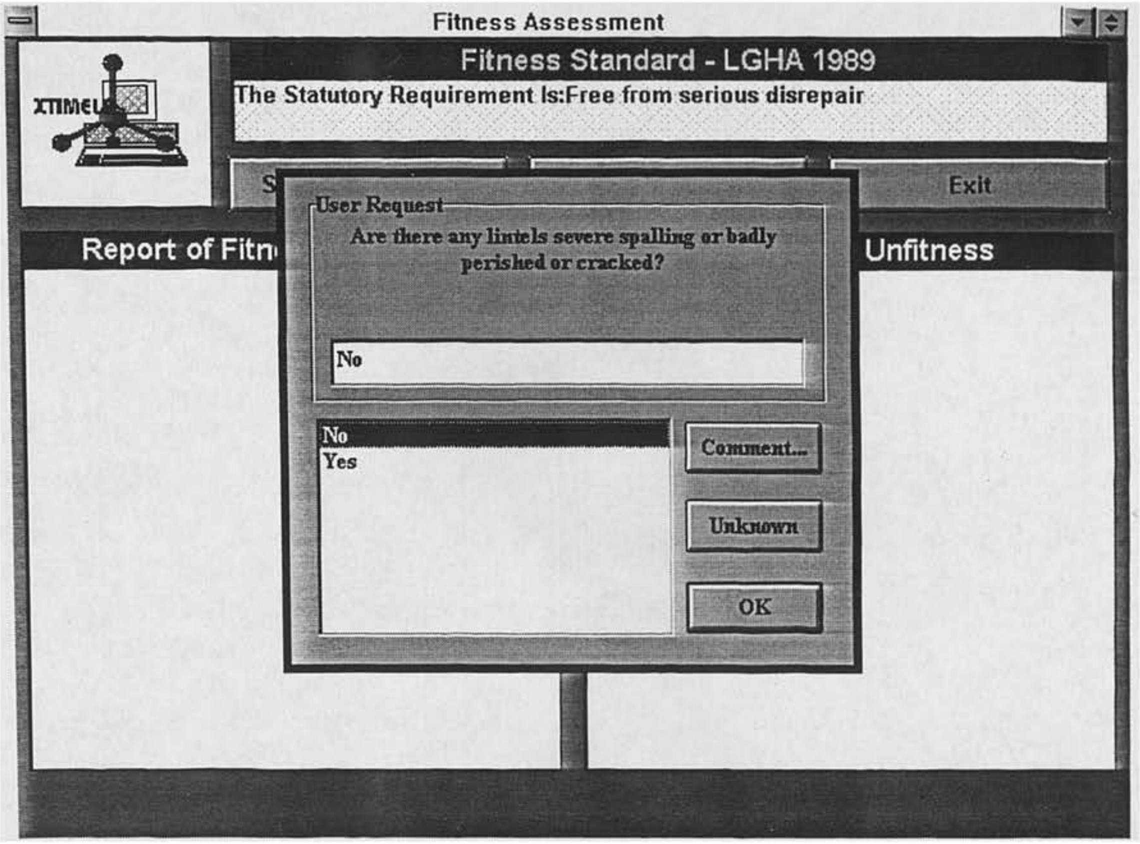
Screen 48: Input information about the condition of the building components



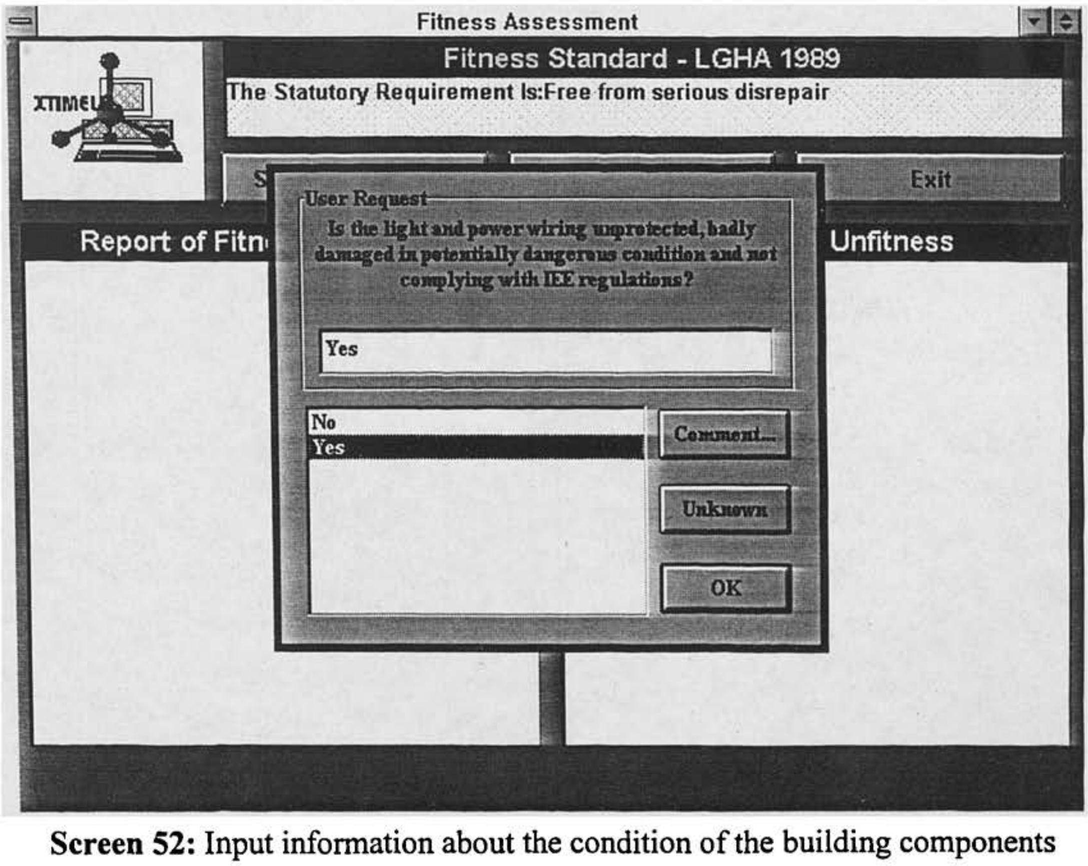
Screen 49: Input information about the condition of the building components



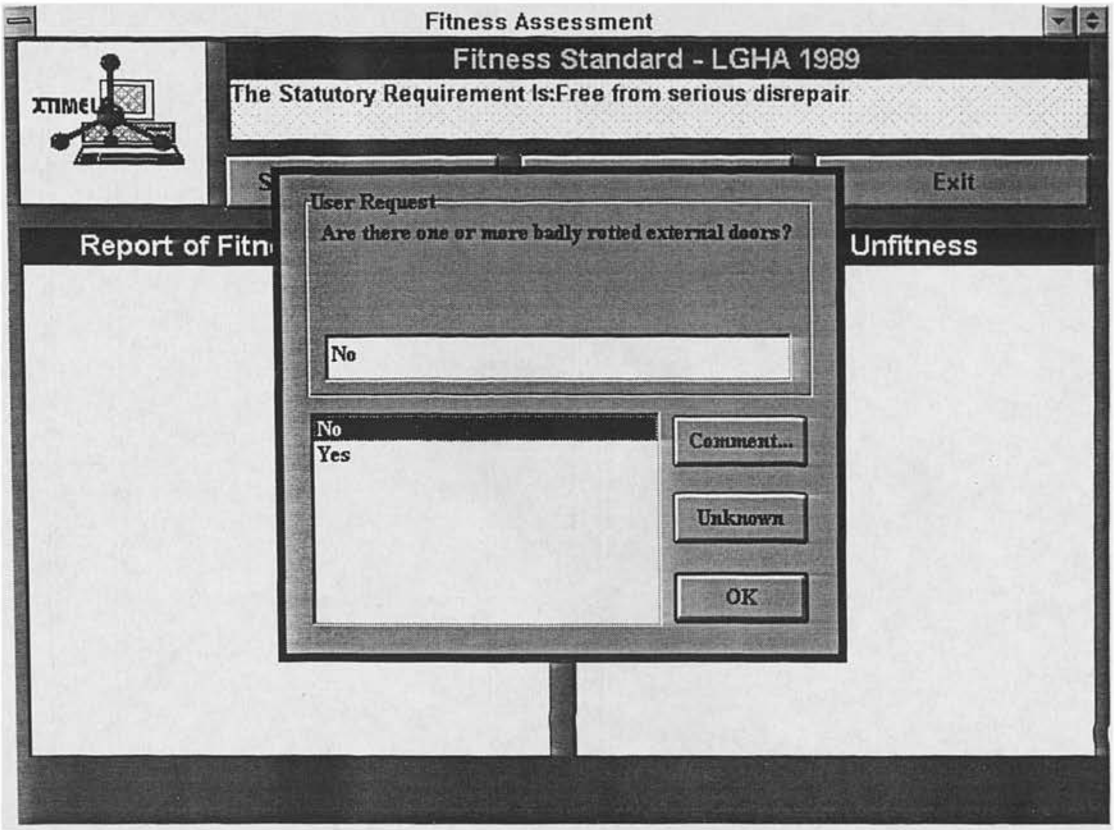
Screen 50: Input information about the condition of the building components



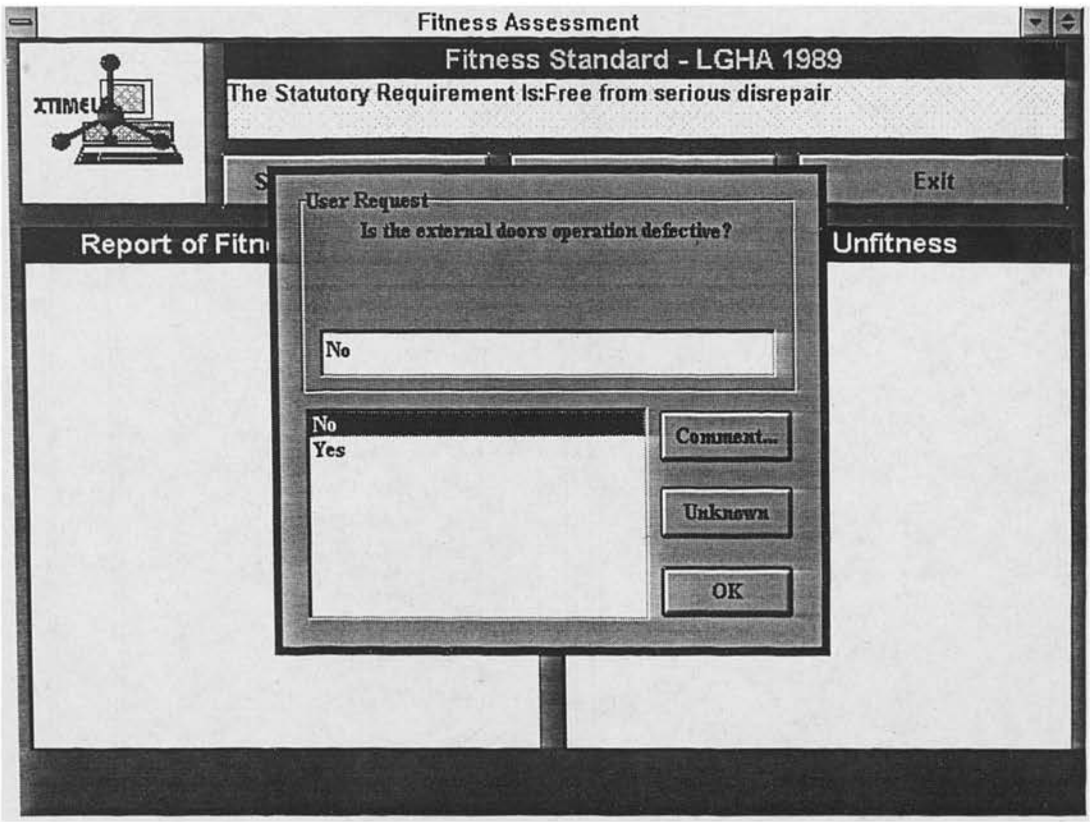
Screen 51: Input information about the condition of the building components



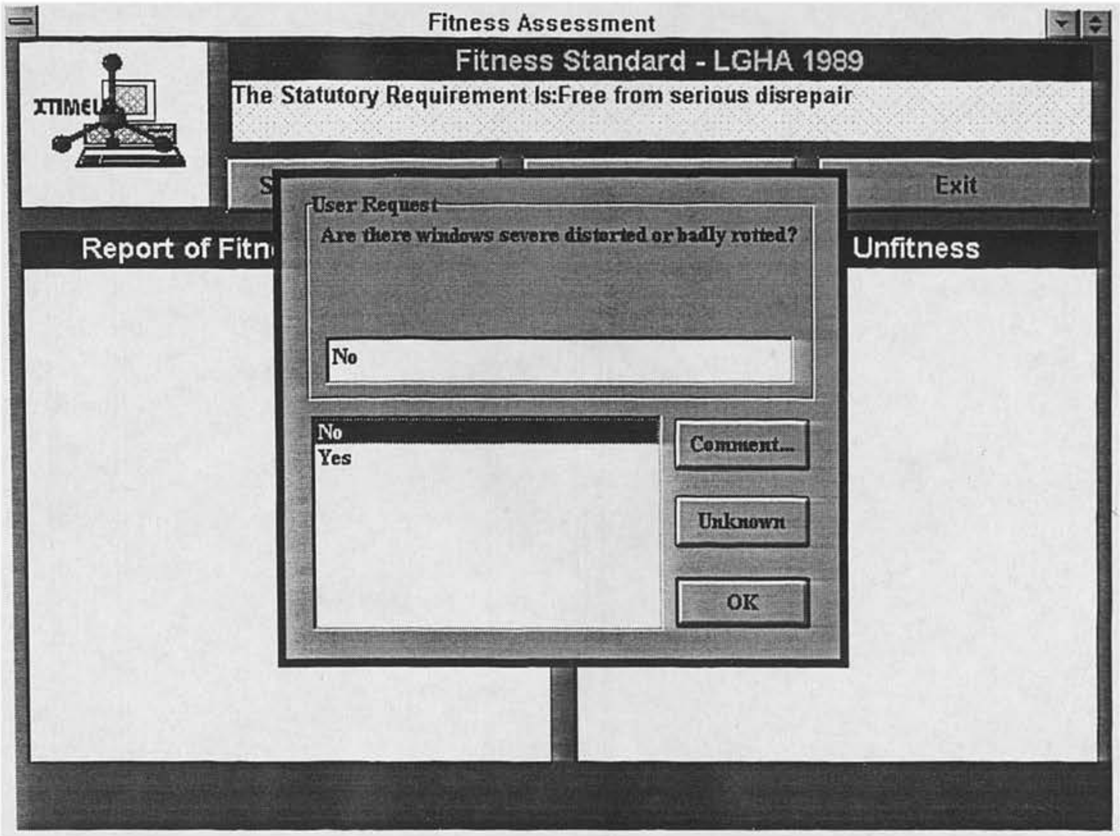
Screen 52: Input information about the condition of the building components



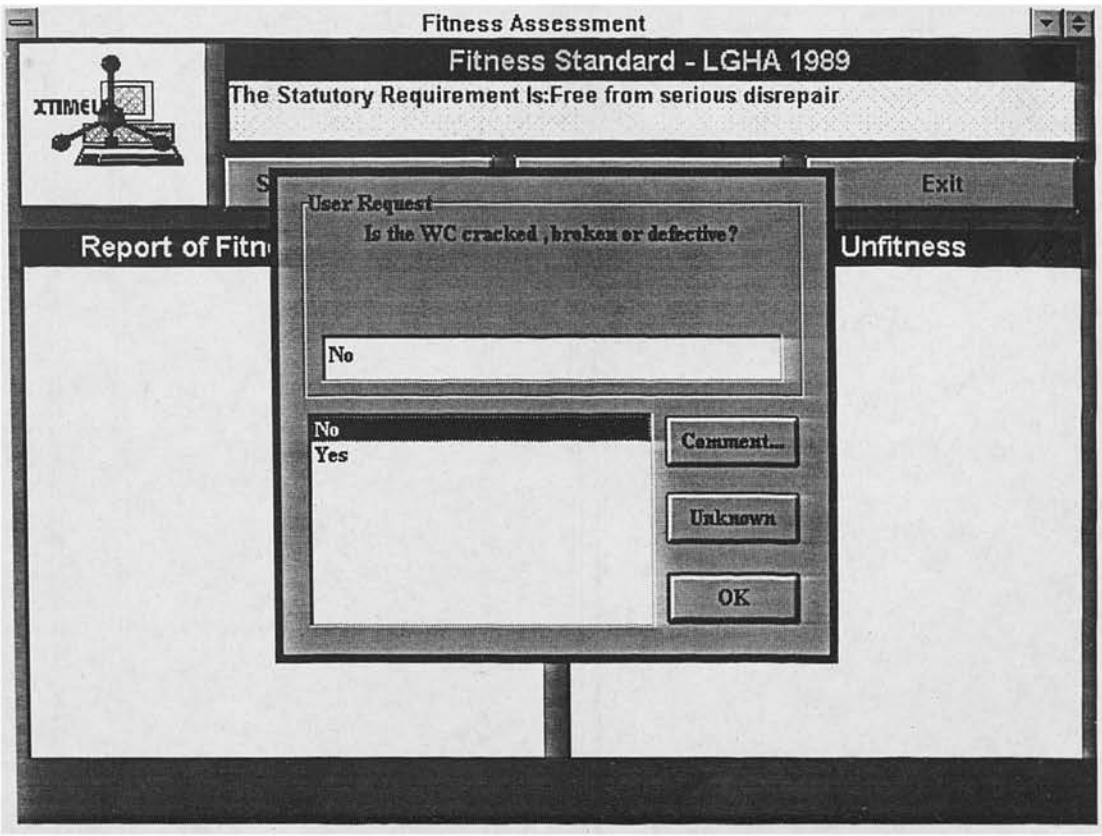
Screen 53: Input information about the condition of the building components



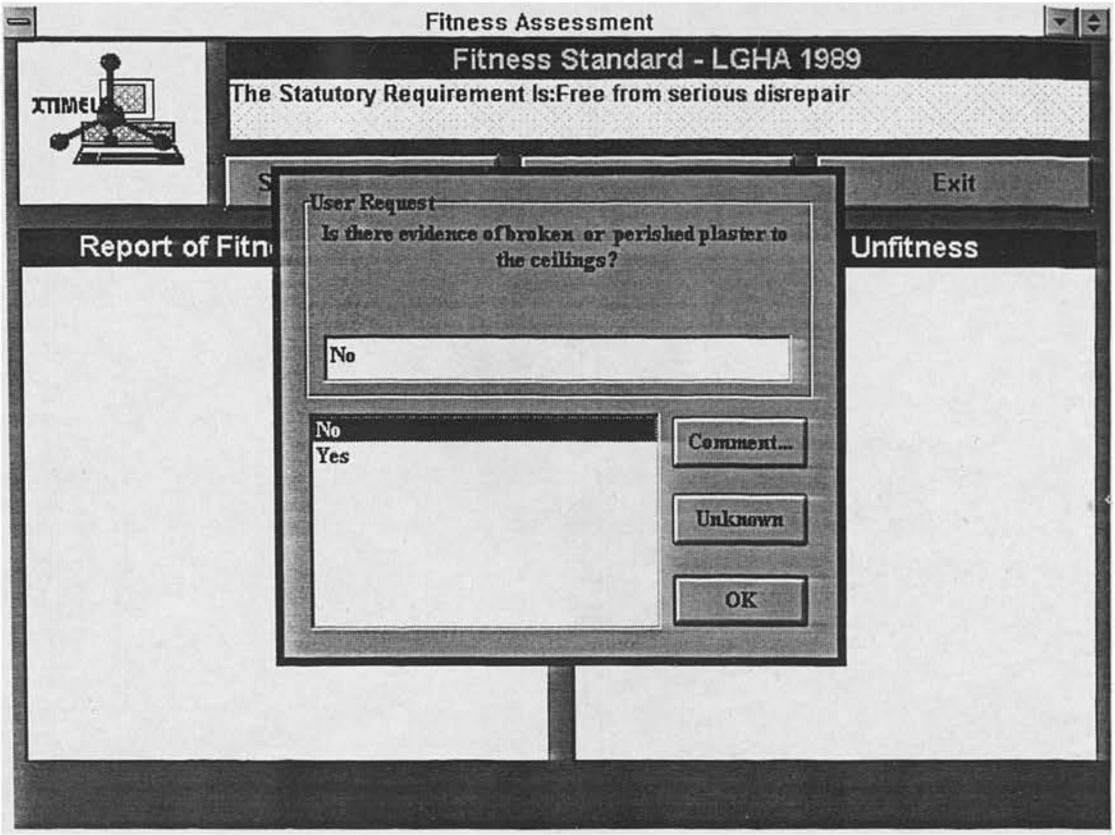
Screen 54: Input information about the condition of the building components



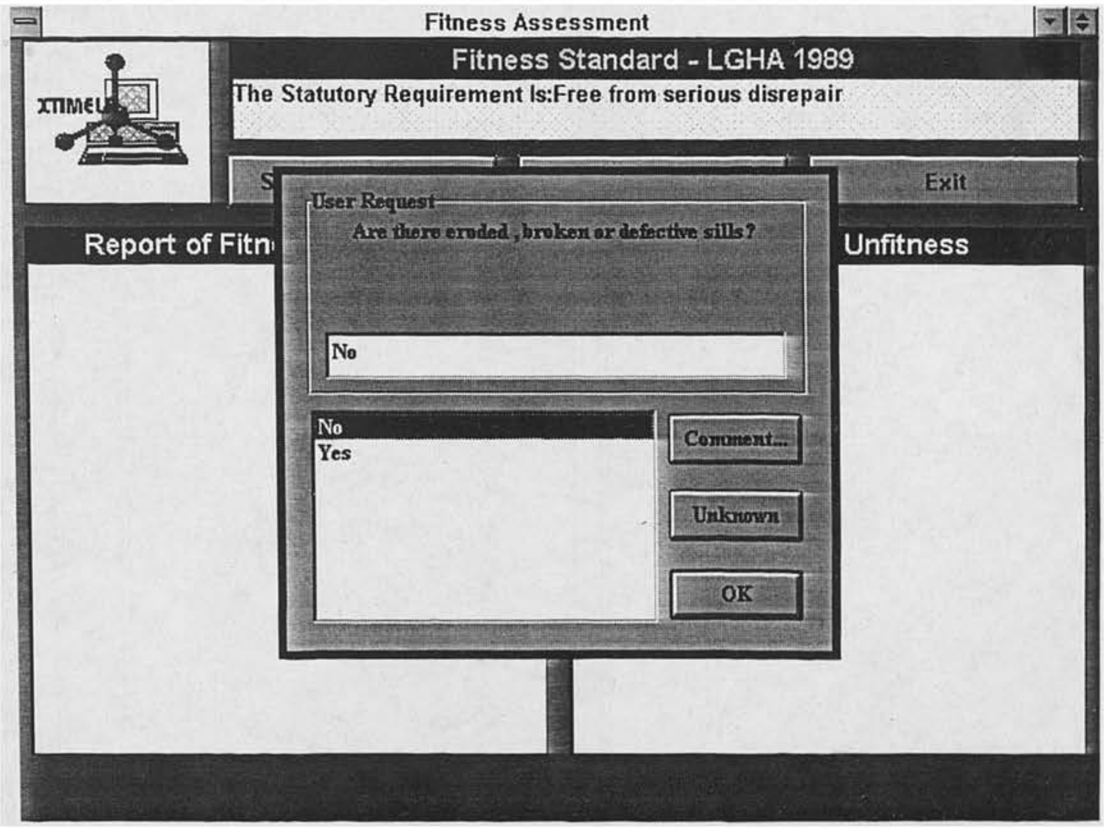
Screen 55: Input information about the condition of the building components



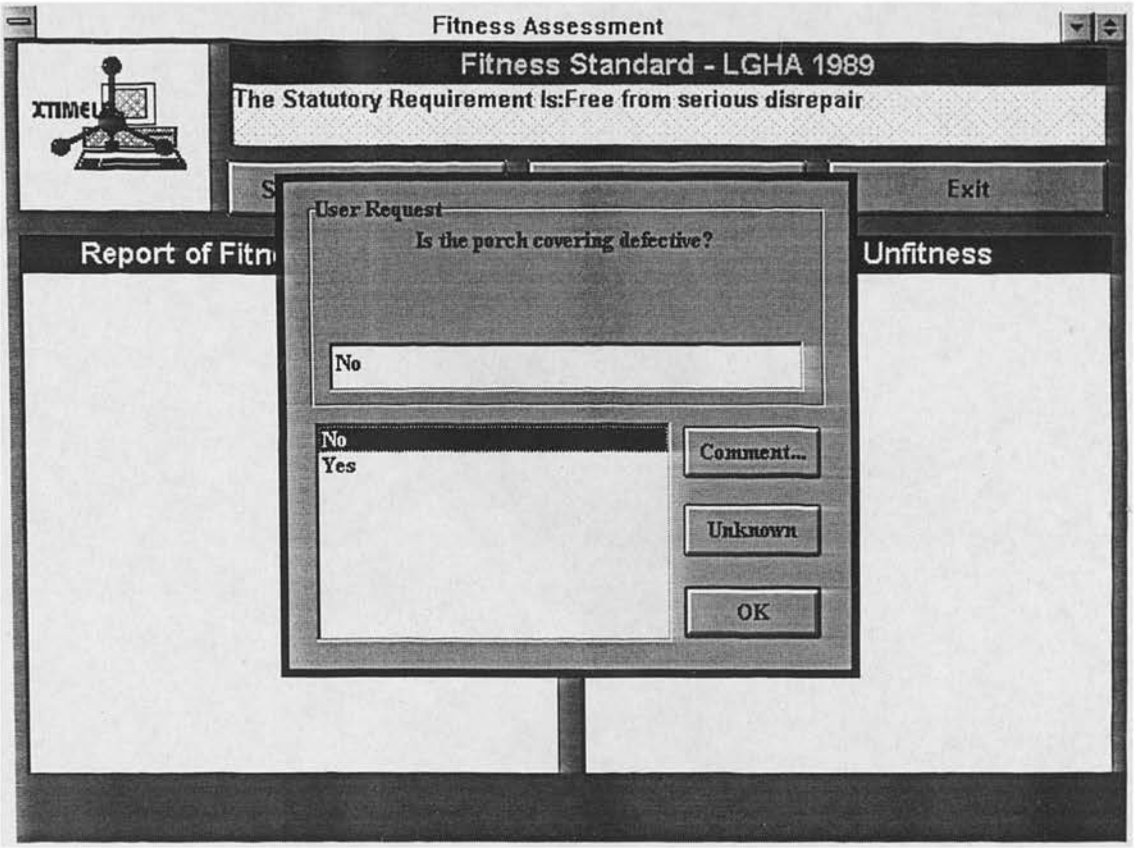
Screen 56: Input information about the condition of the building components



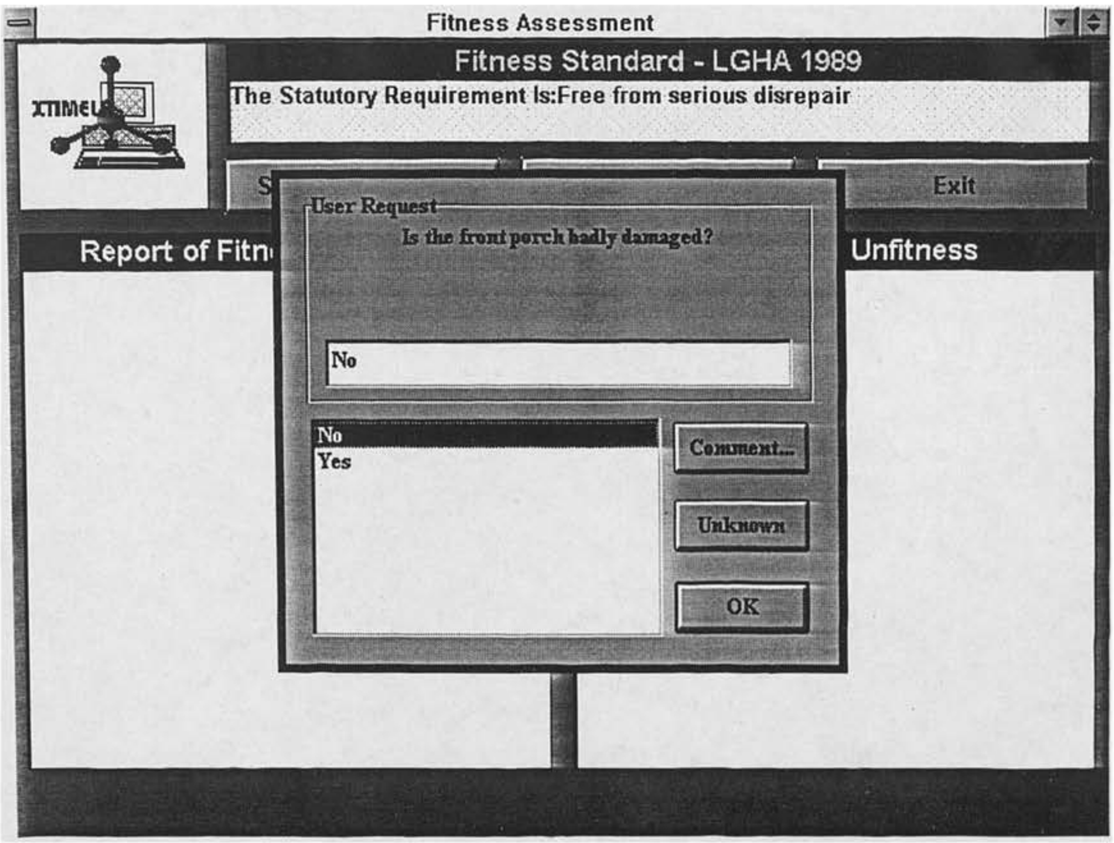
Screen 57: Input information about the condition of the building components



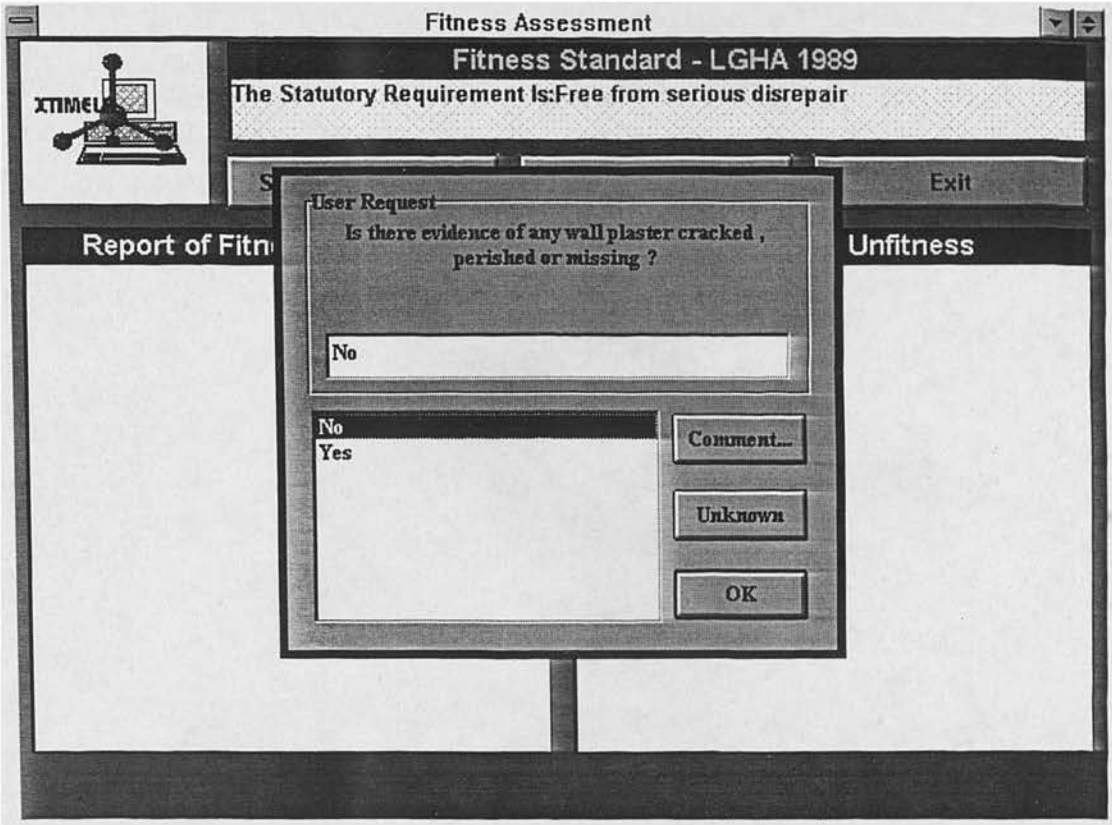
Screen 58: Input information about the condition of the building components



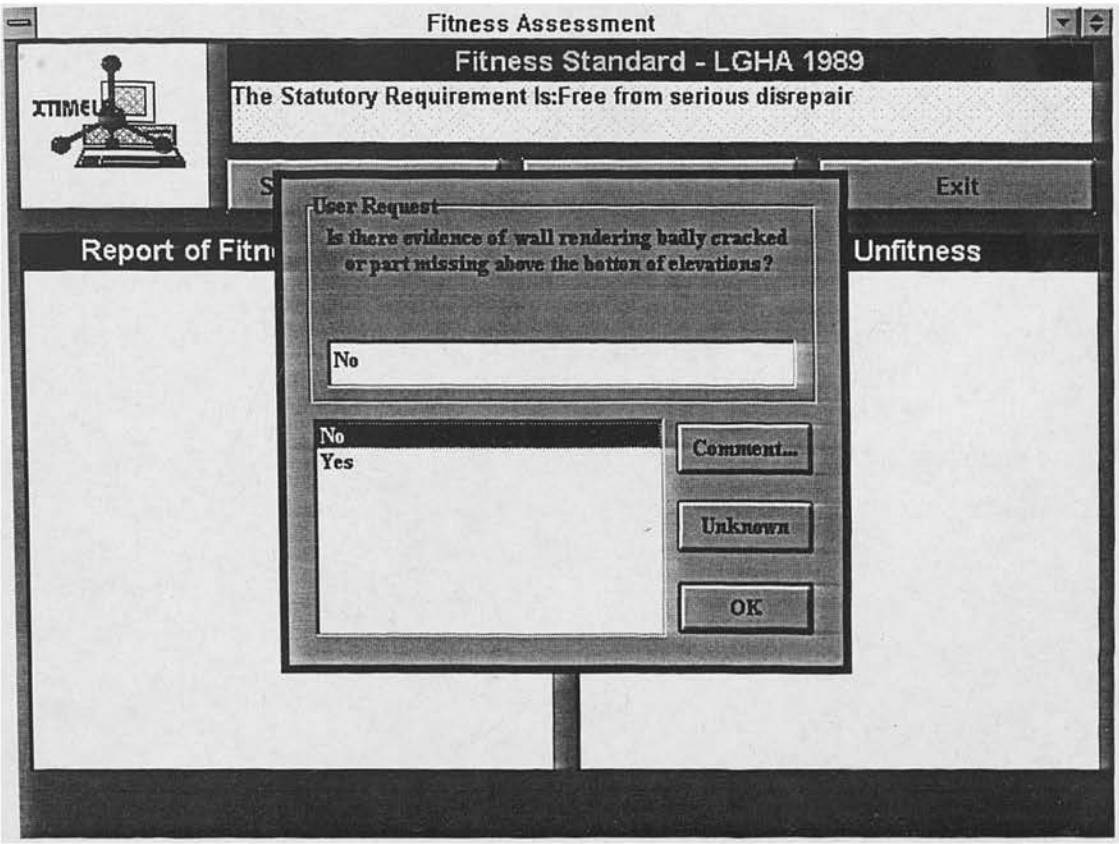
Screen 59: Input information about the condition of the building components



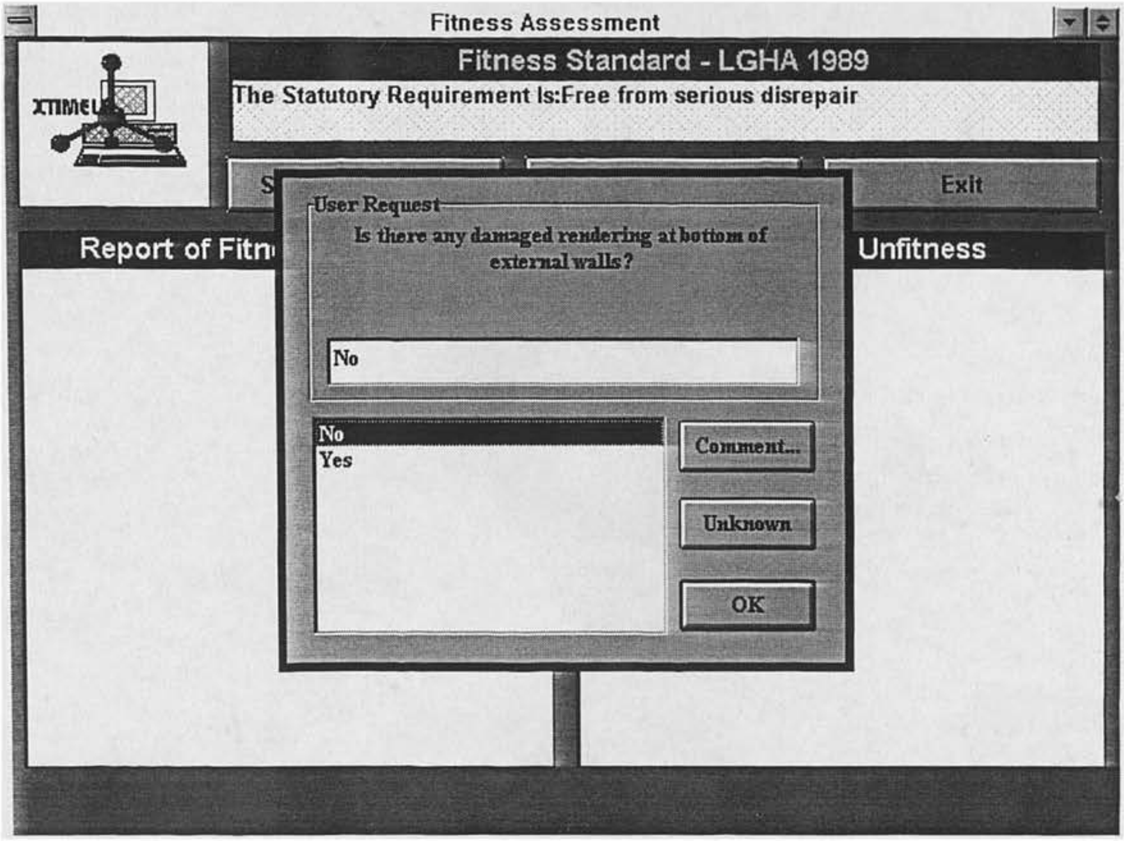
Screen 60: Input information about the condition of the building components



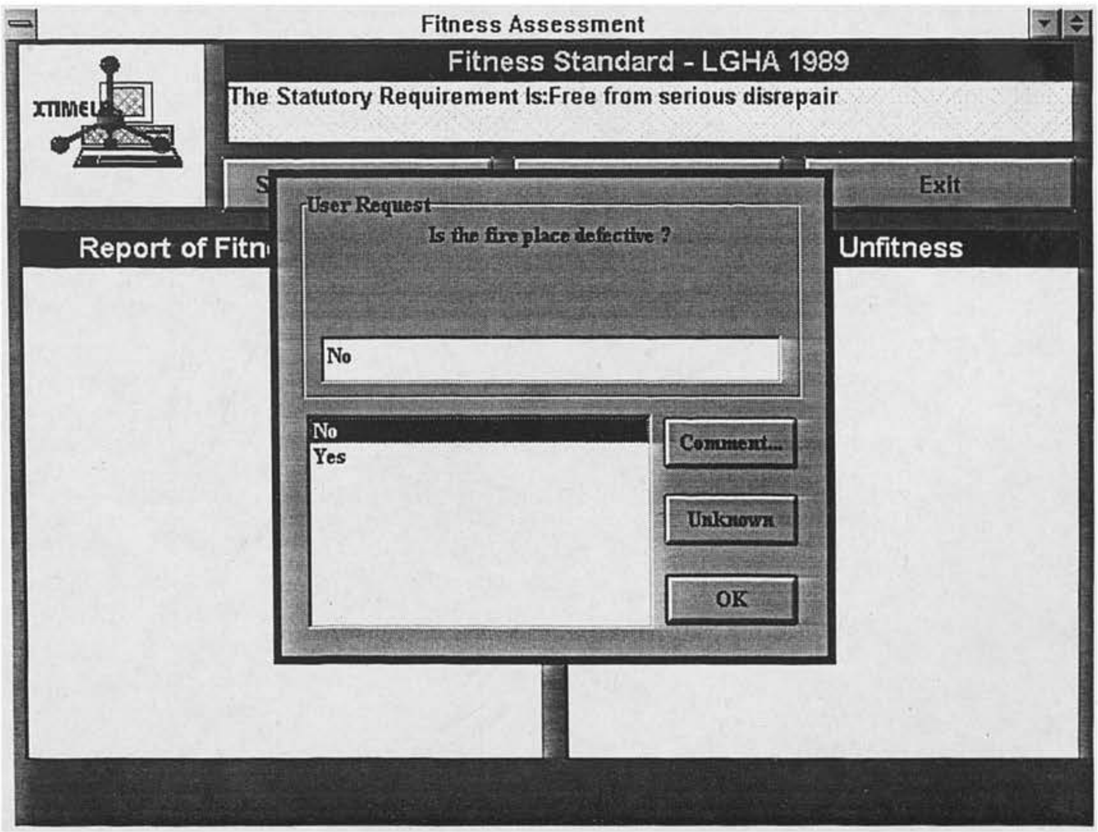
Screen 61: Input information about the condition of the building components



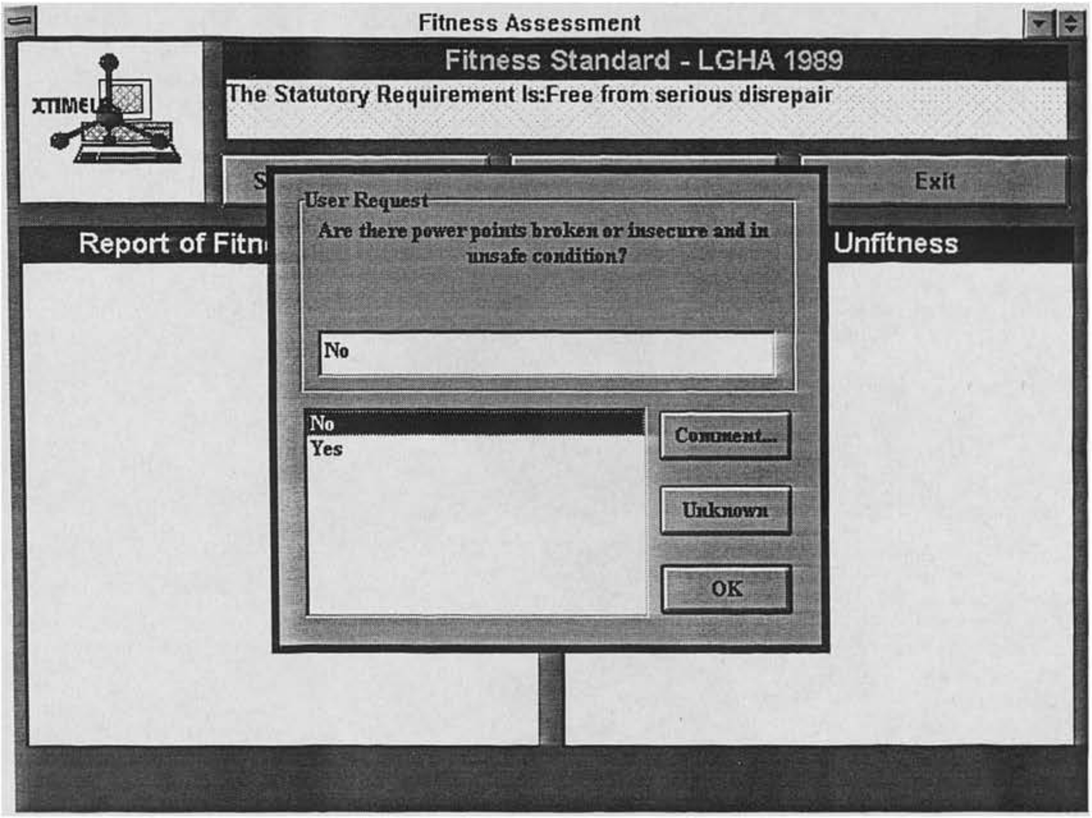
Screen 62: Input information about the condition of the building components



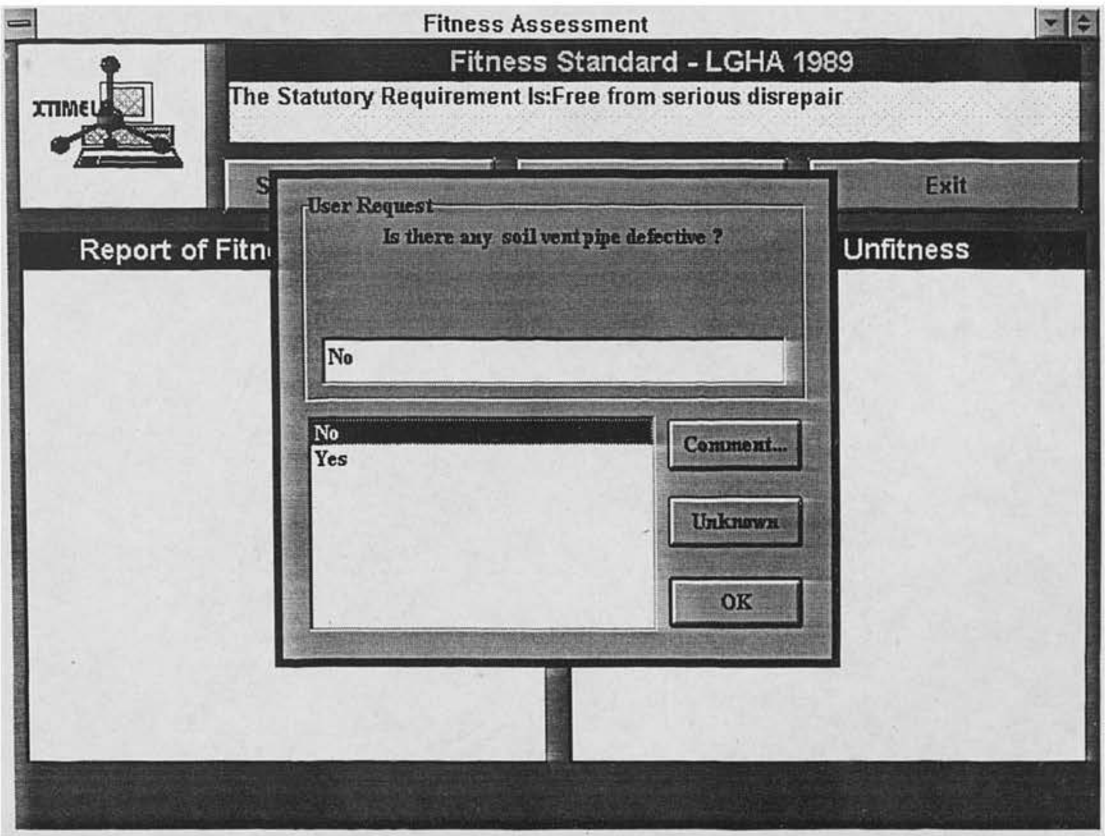
Screen 63: Input information about the condition of the building components



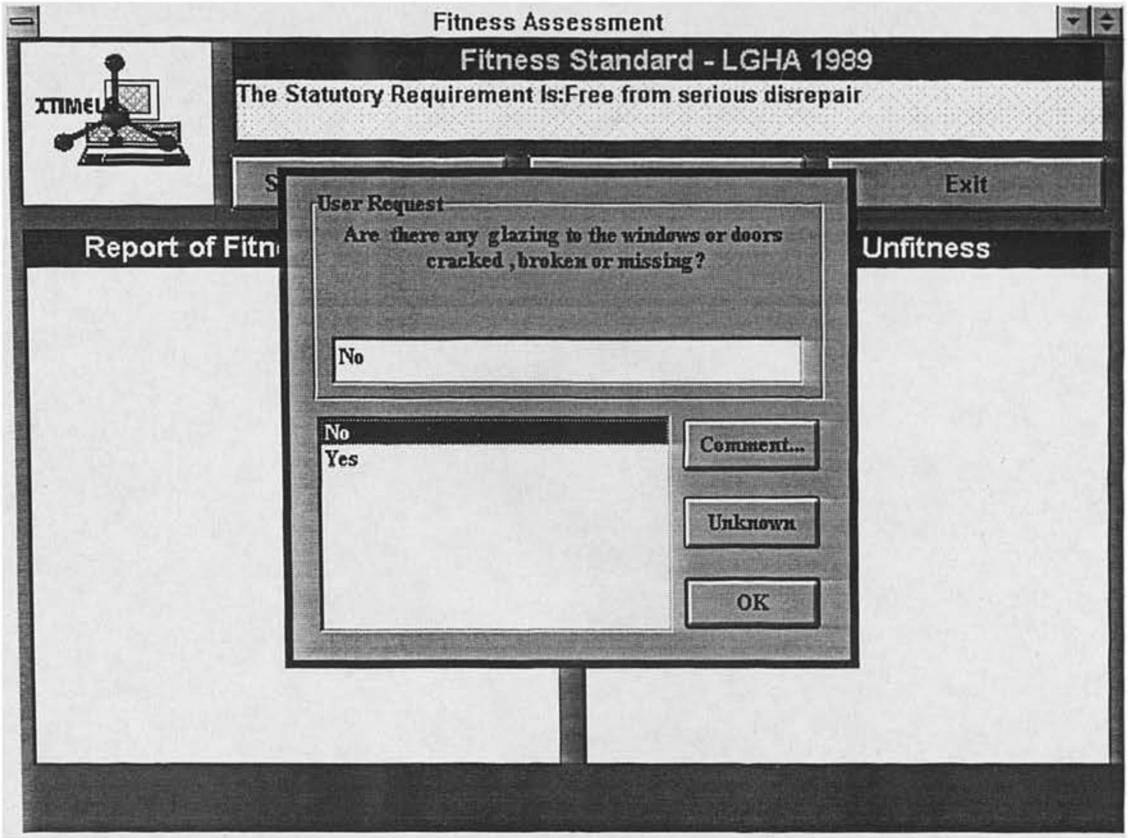
Screen 64: Input information about the condition of the building components



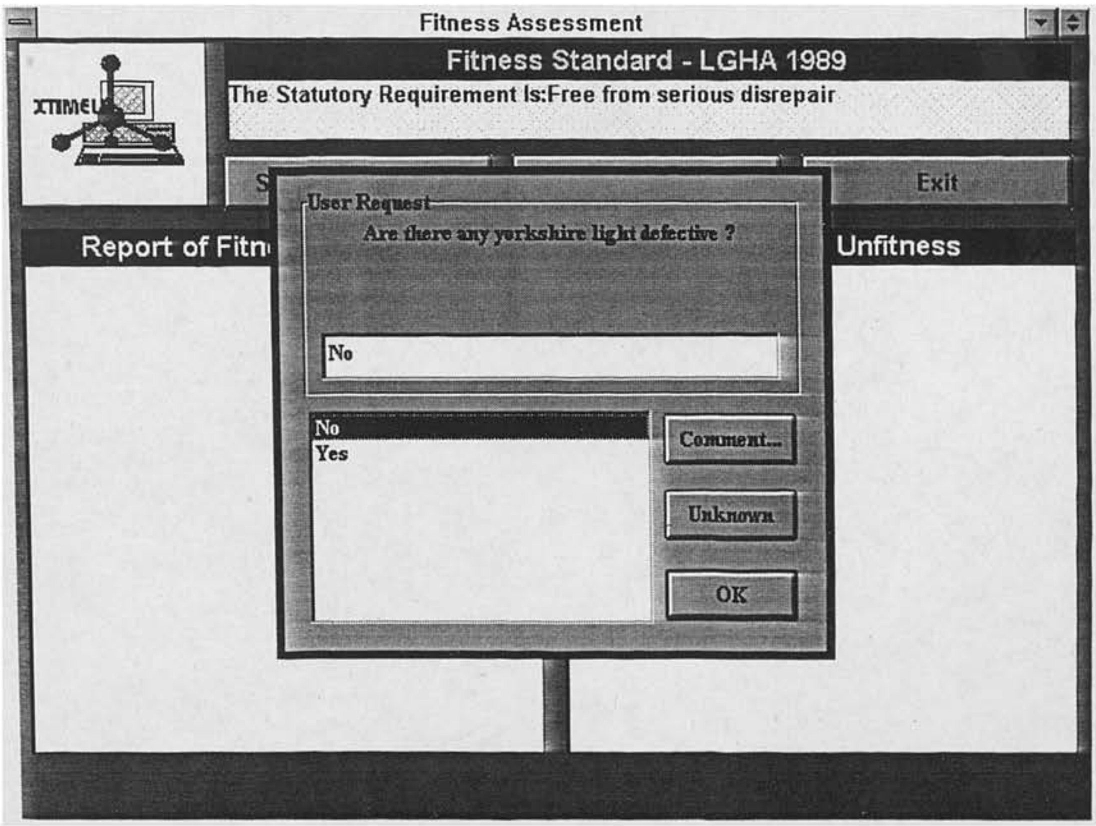
Screen 65: Input information about the condition of the building components



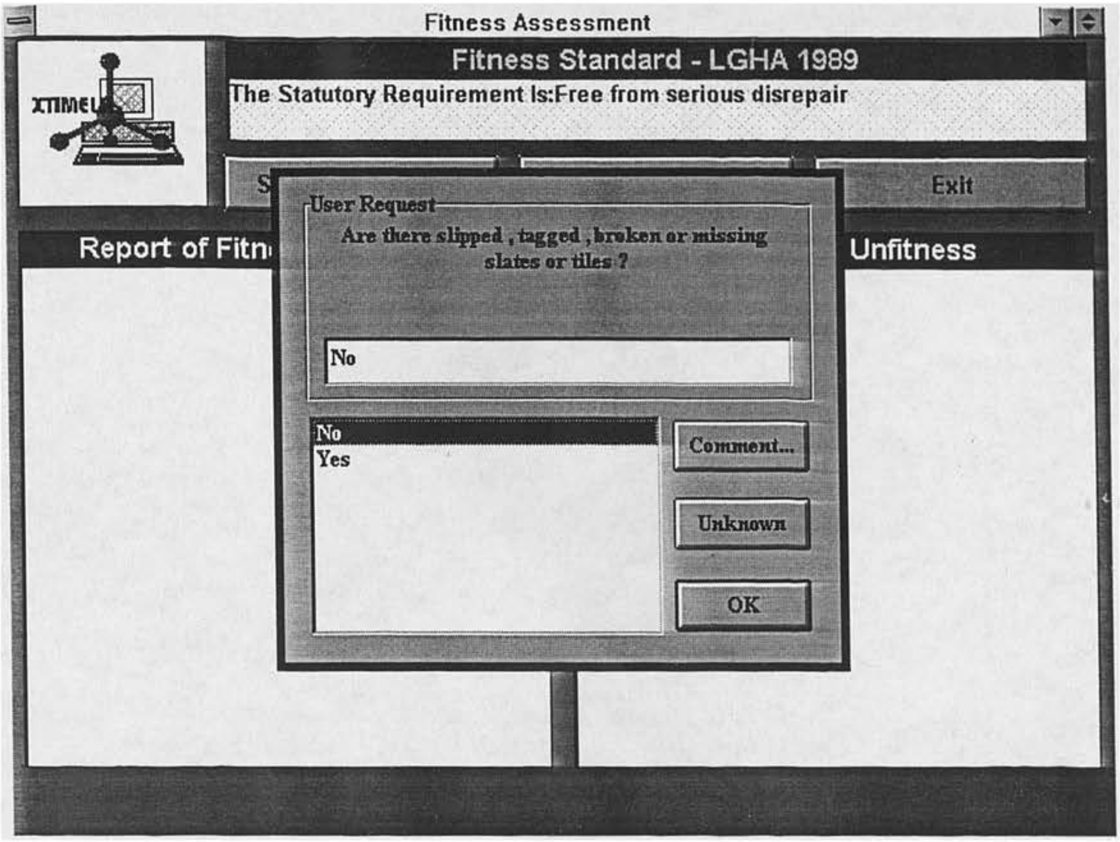
Screen 66: Input information about the condition of the building components



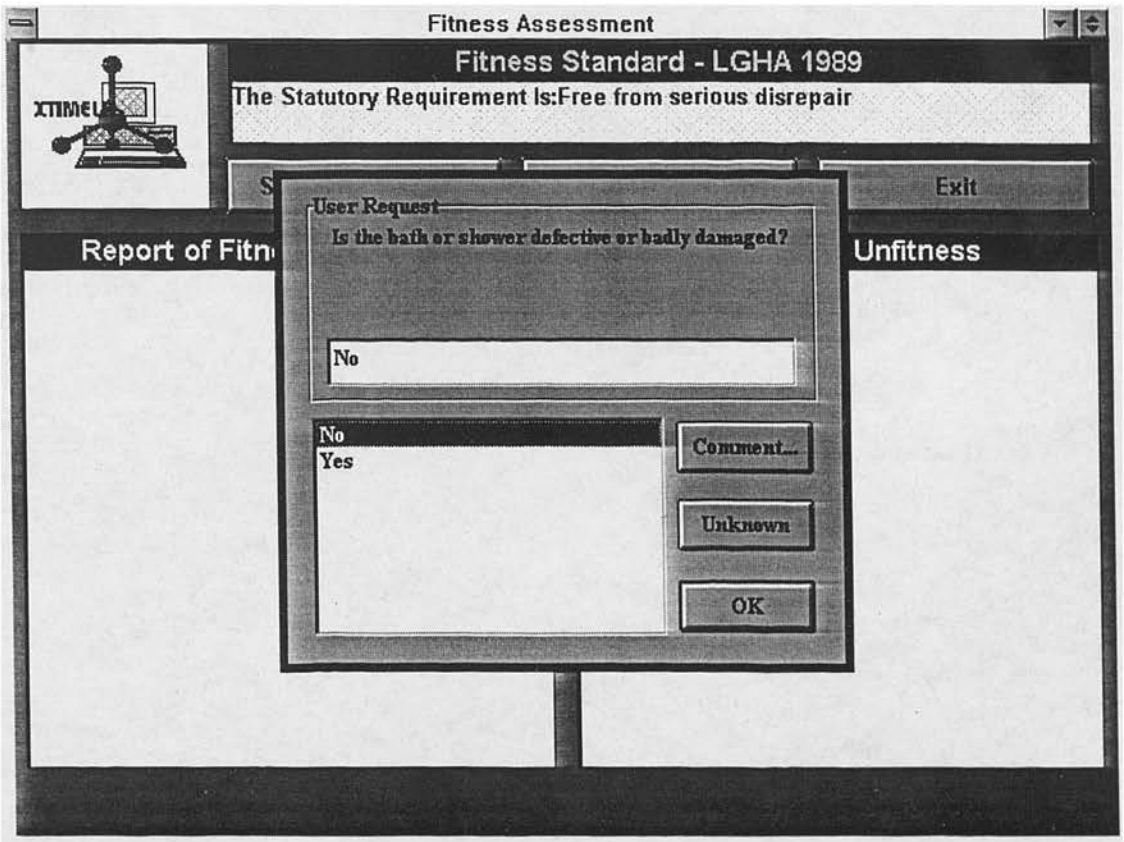
Screen 67: Input information about the condition of the building components



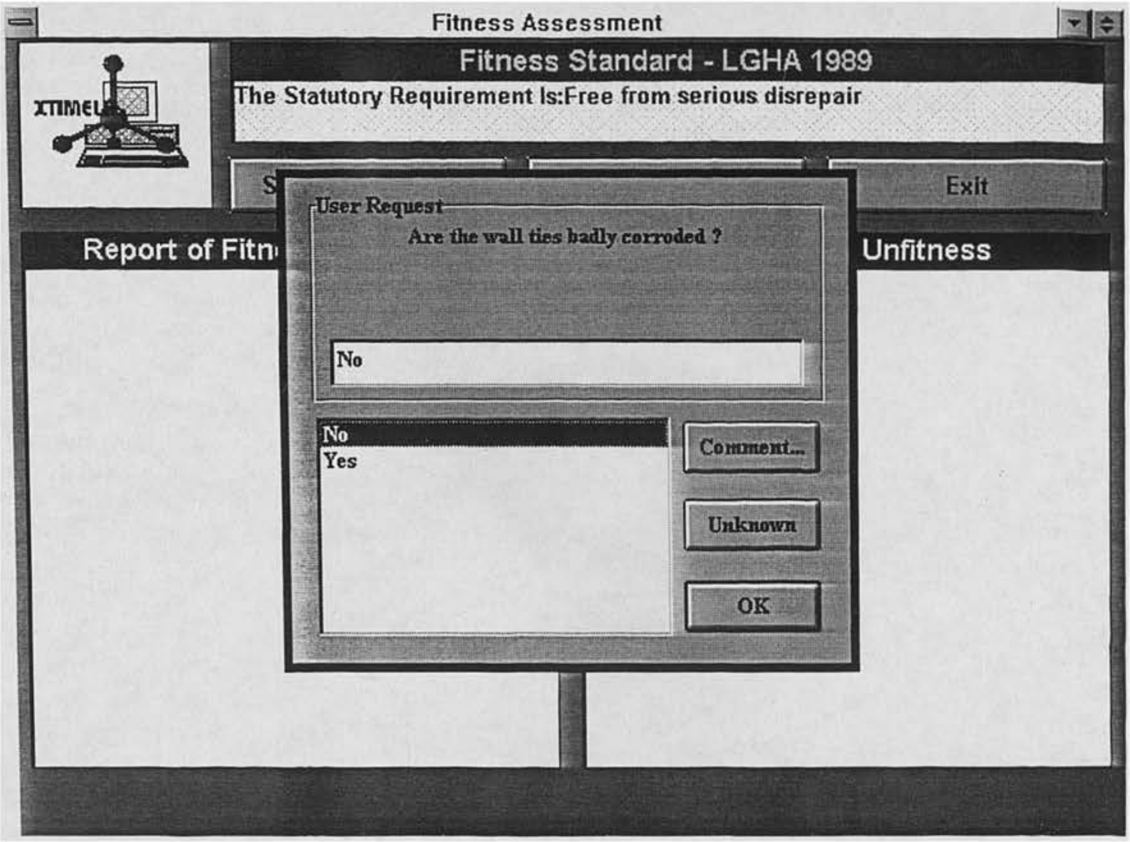
Screen 68: Input information about the condition of the building components



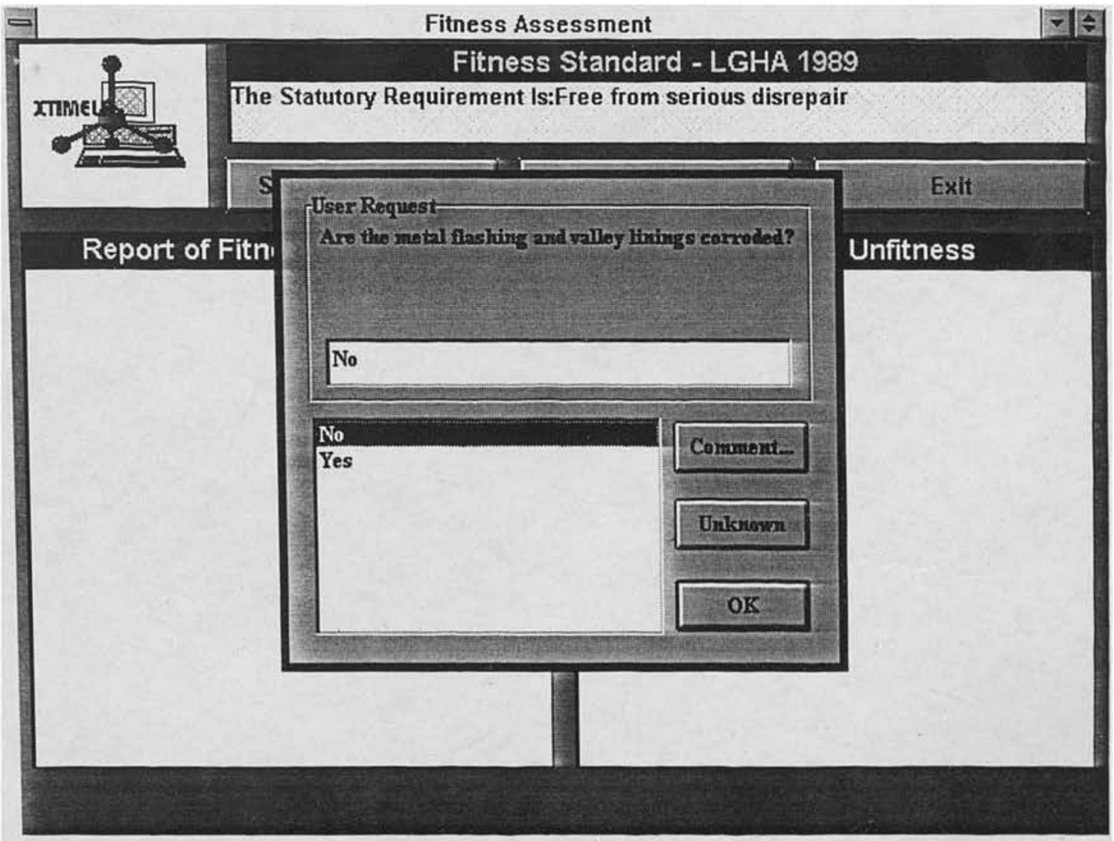
Screen 69: Input information about the condition of the building components



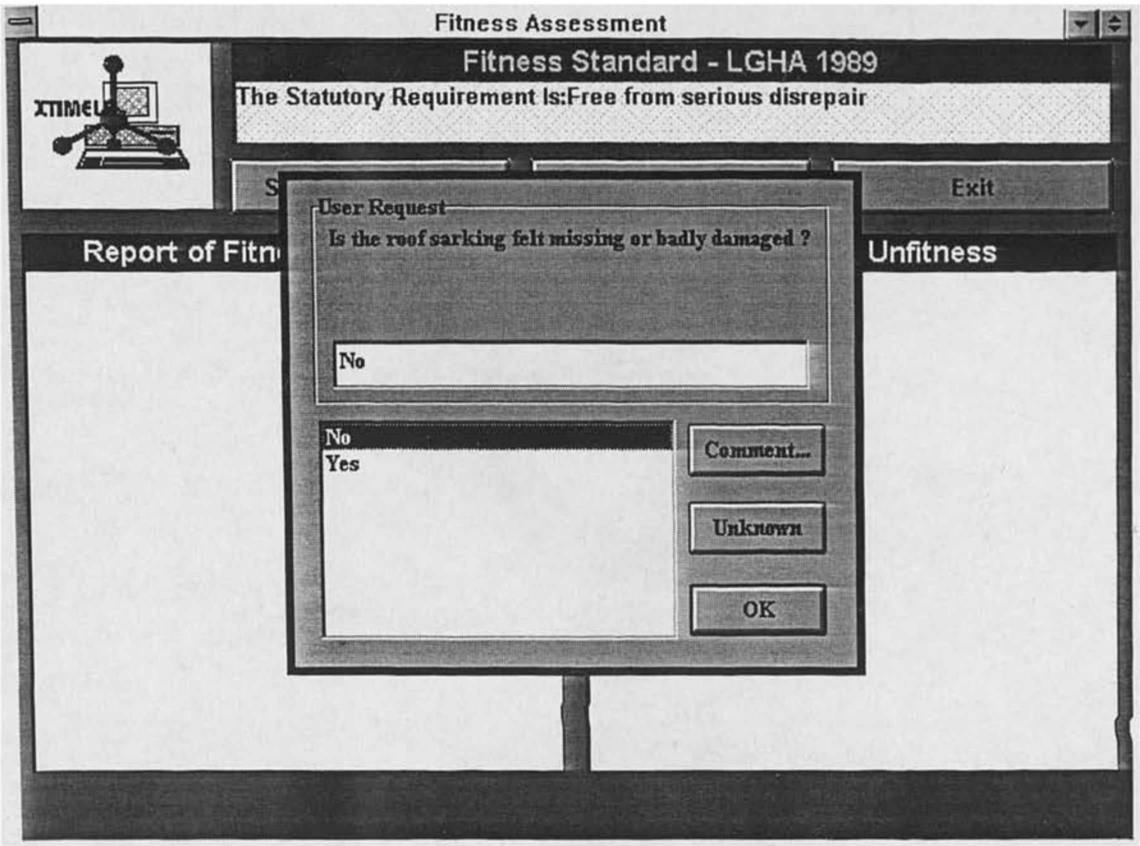
Screen 70: Input information about the condition of the building components



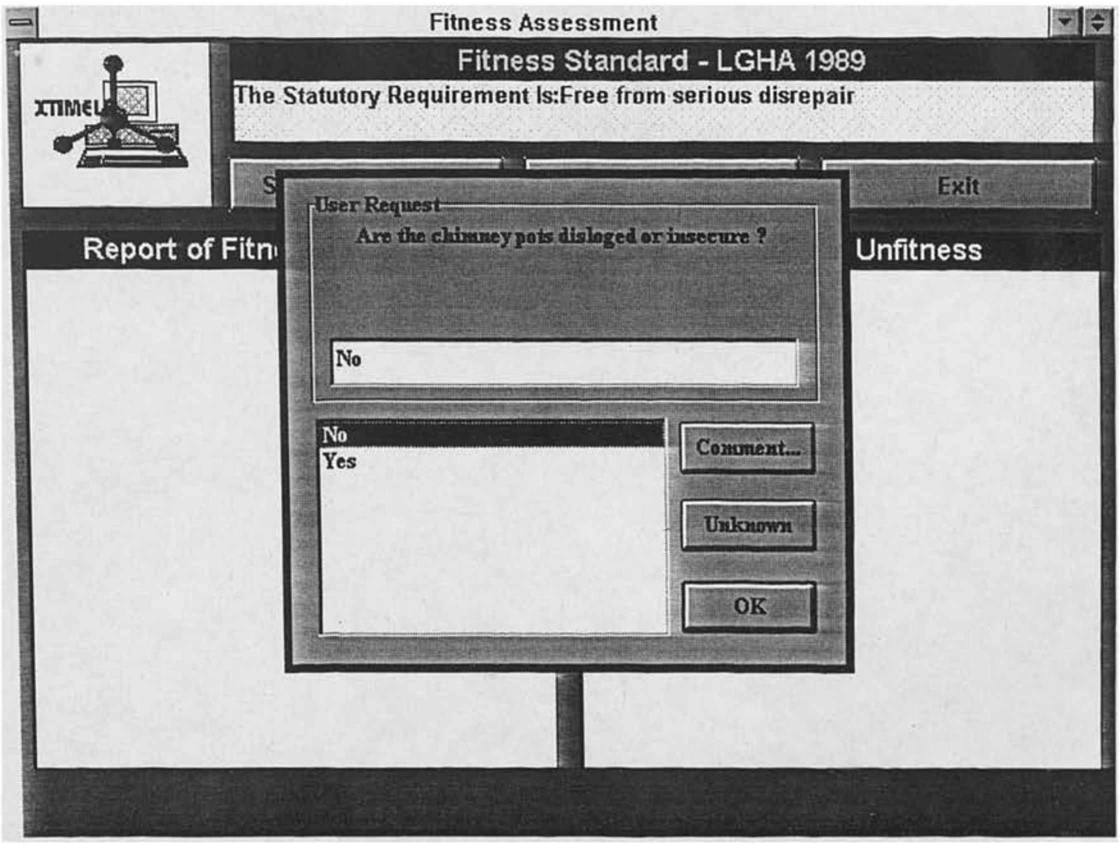
Screen 71: Input information about the condition of the building components



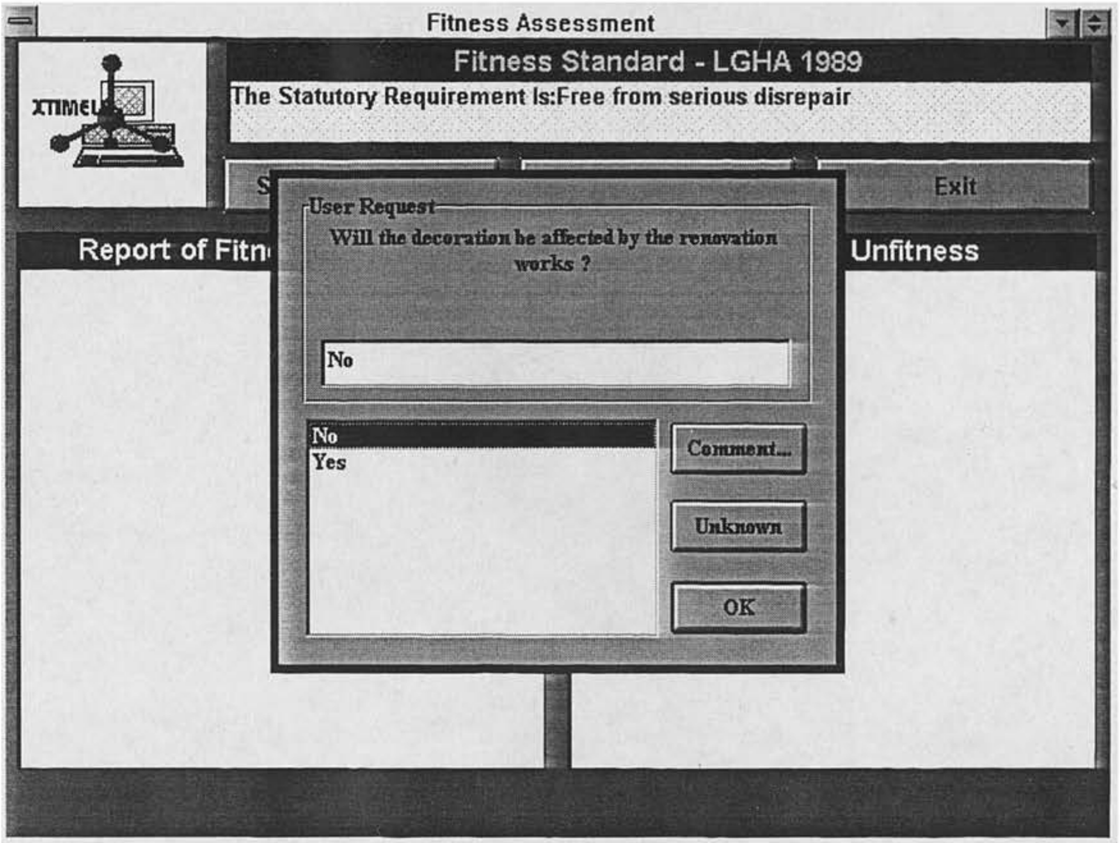
Screen 72: Input information about the condition of the building components



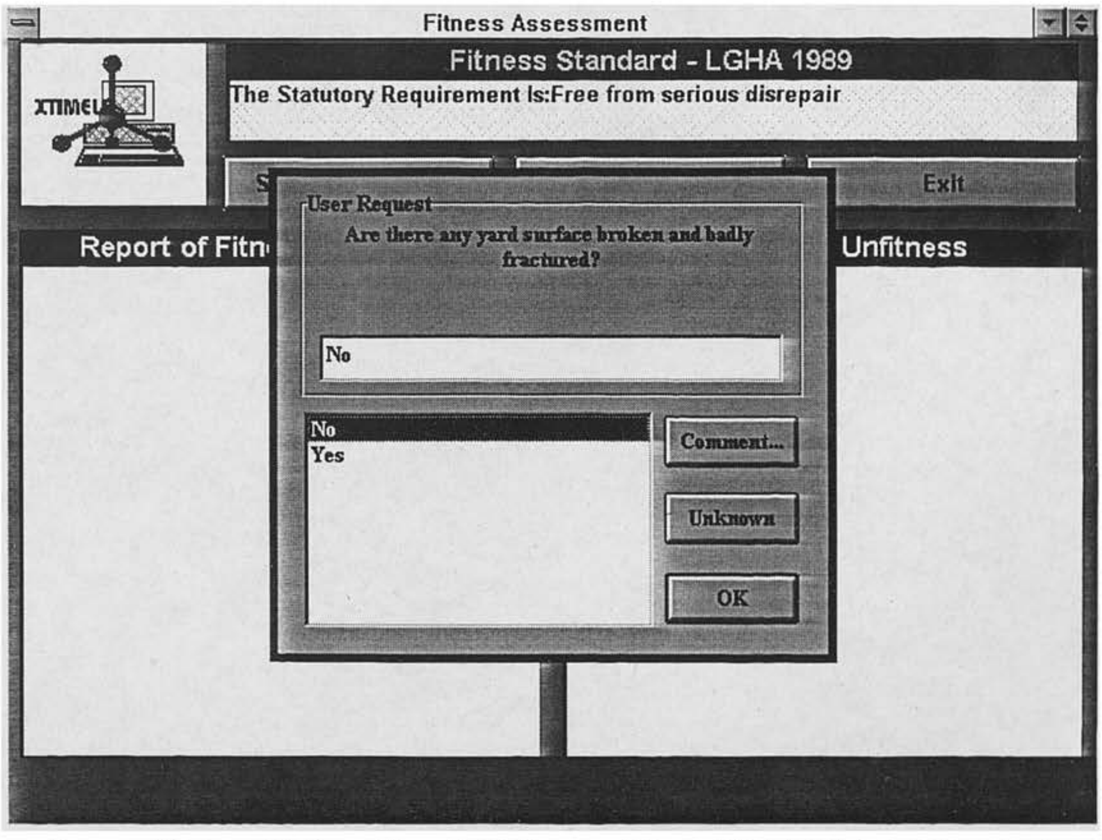
Screen 73: Input information about the condition of the building components



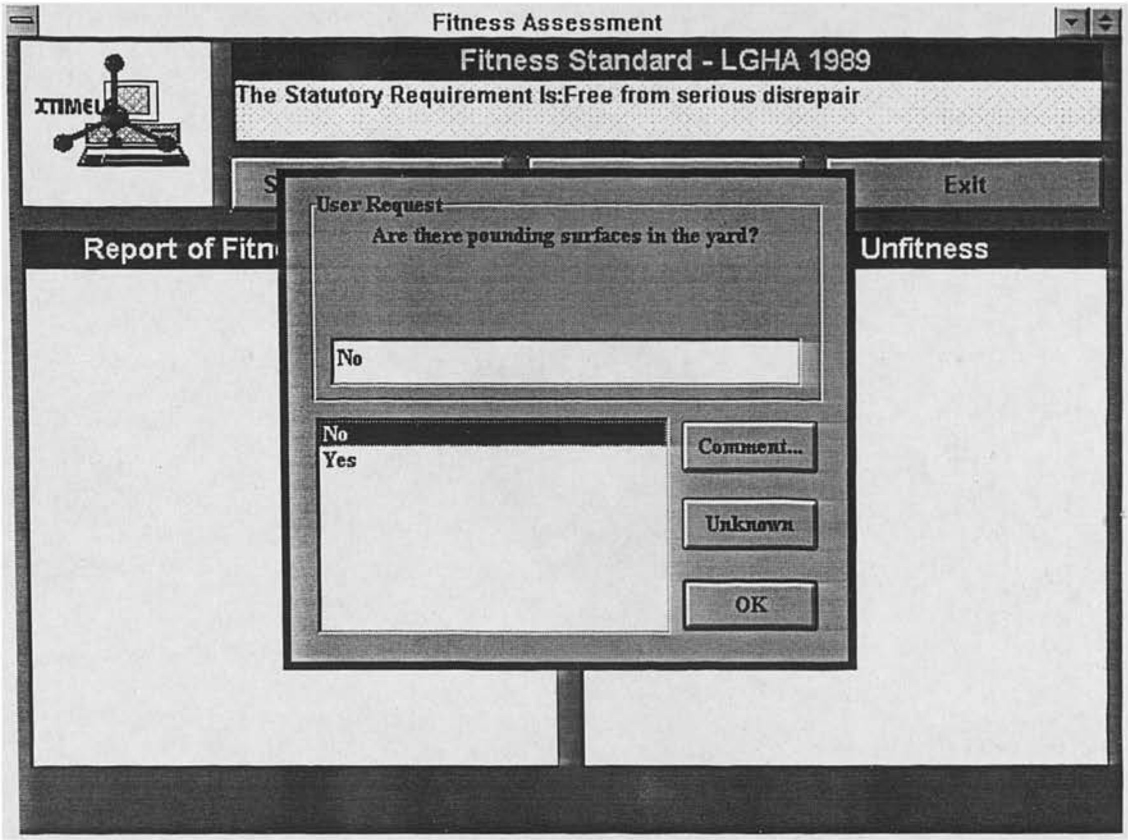
Screen 74: Input information about the condition of the building components



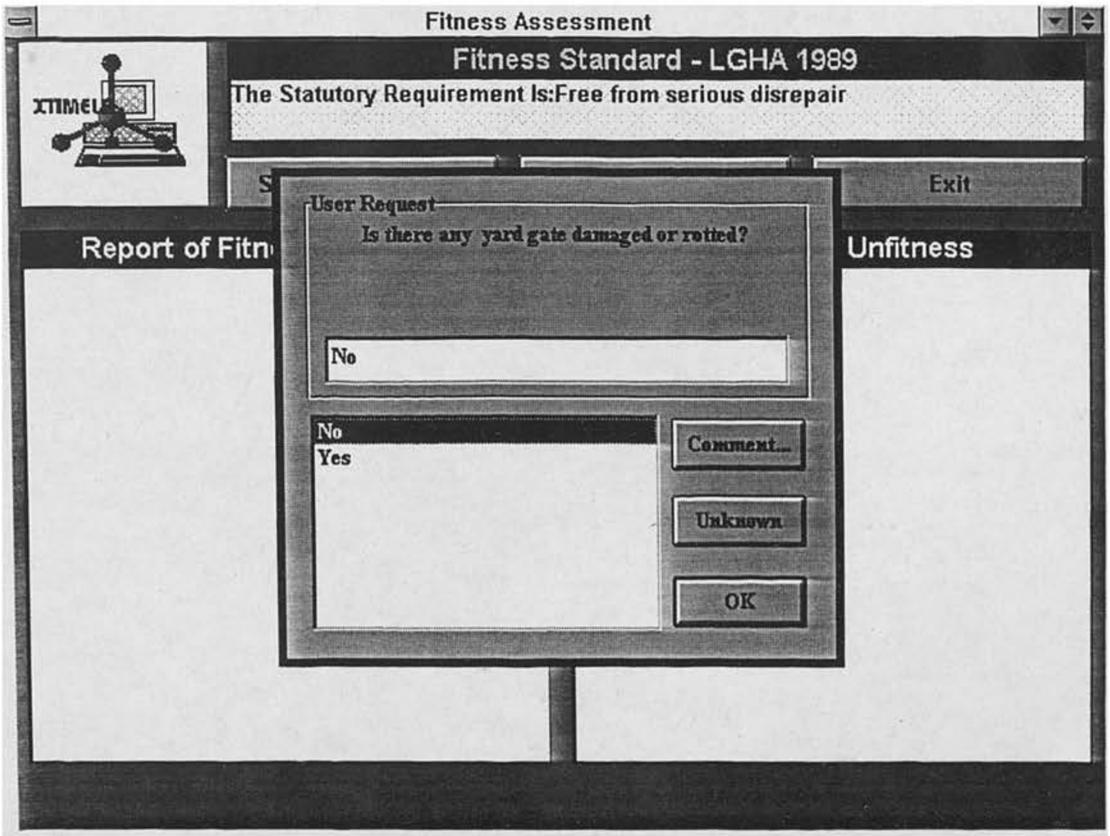
Screen 75: Input information about the condition of the building components



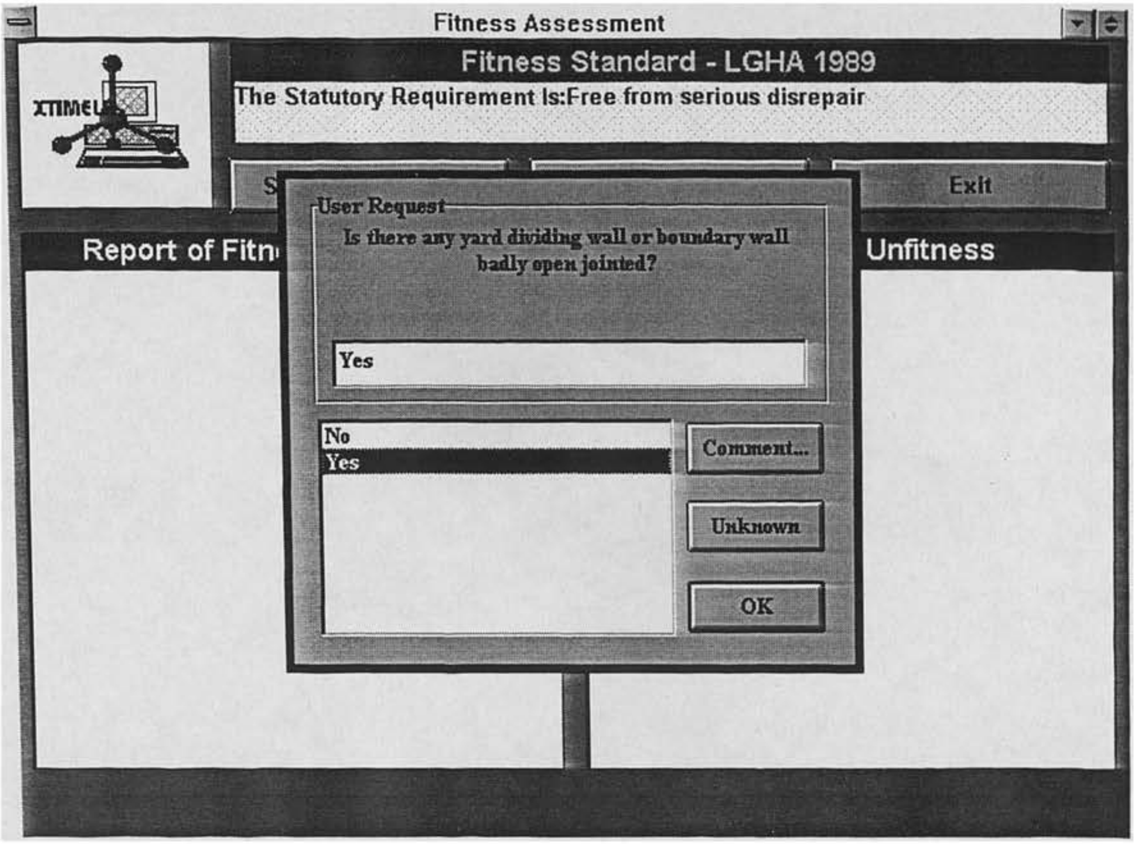
Screen 76: Input information about the condition of the building components



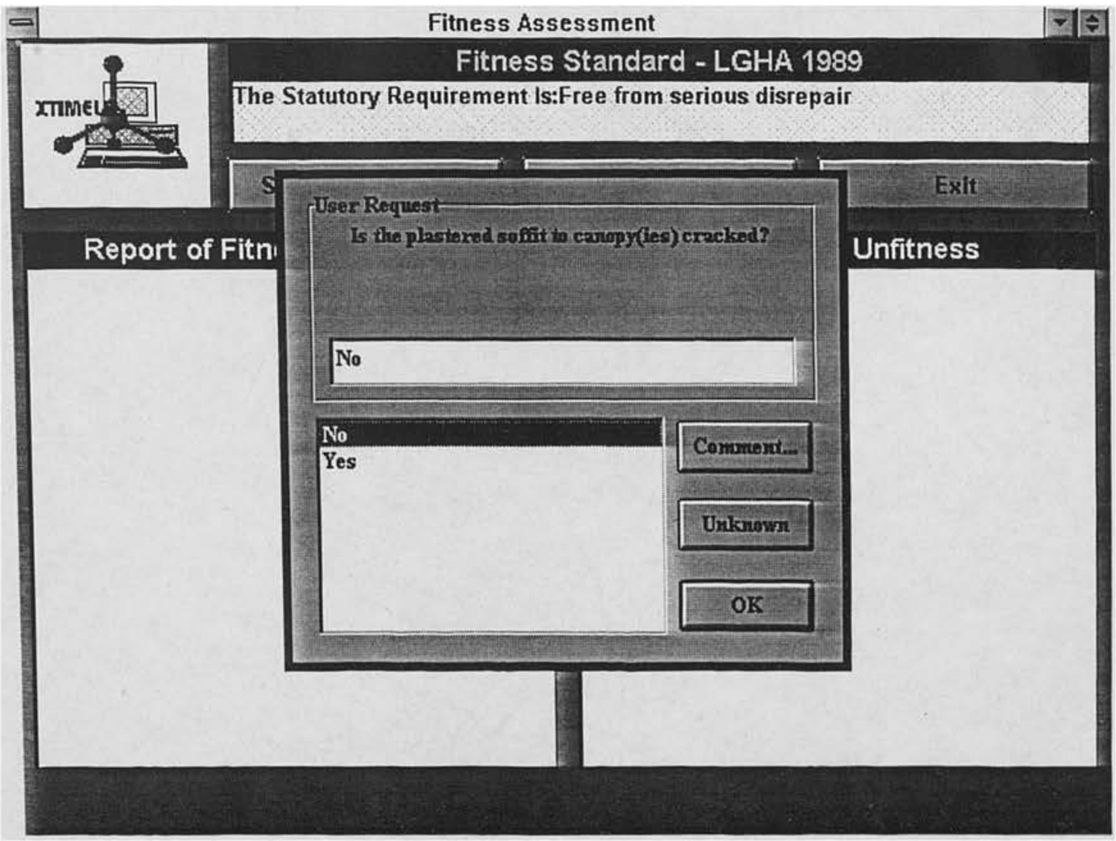
Screen 77: Input information about the condition of the building components



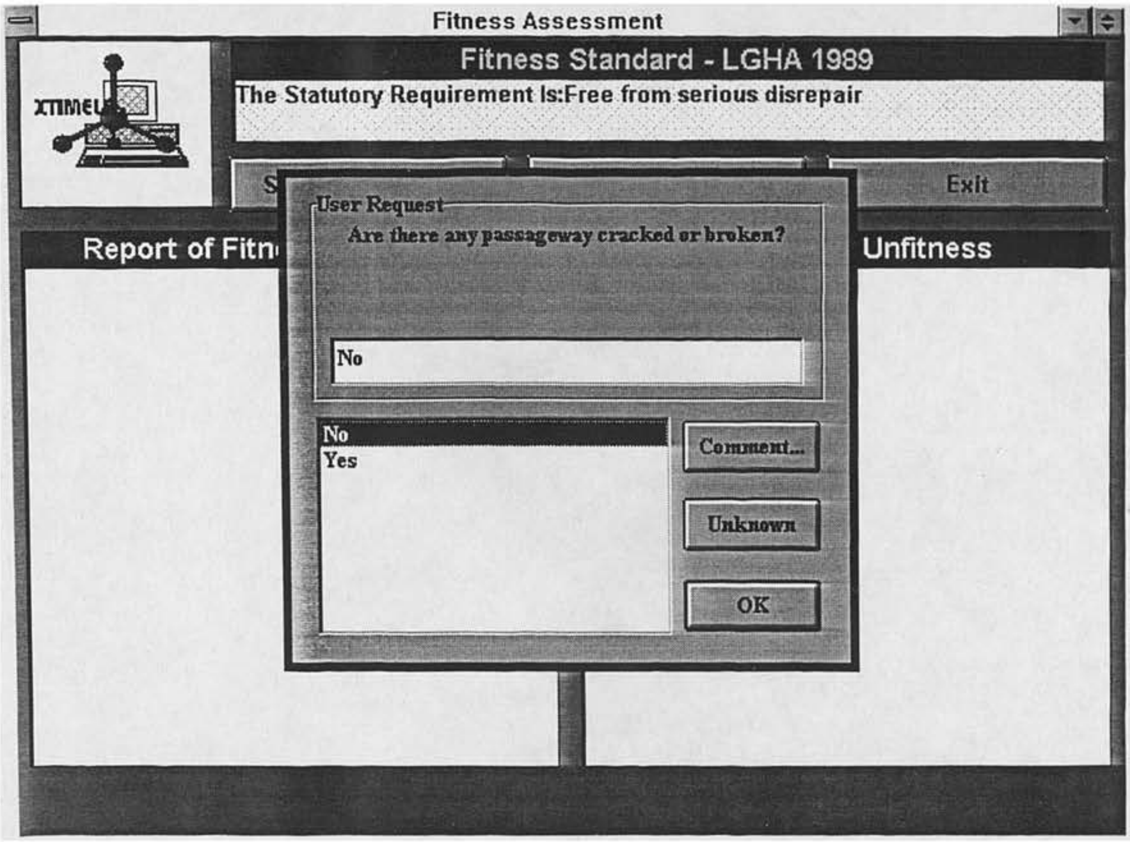
Screen 78: Input information about the condition of the building components



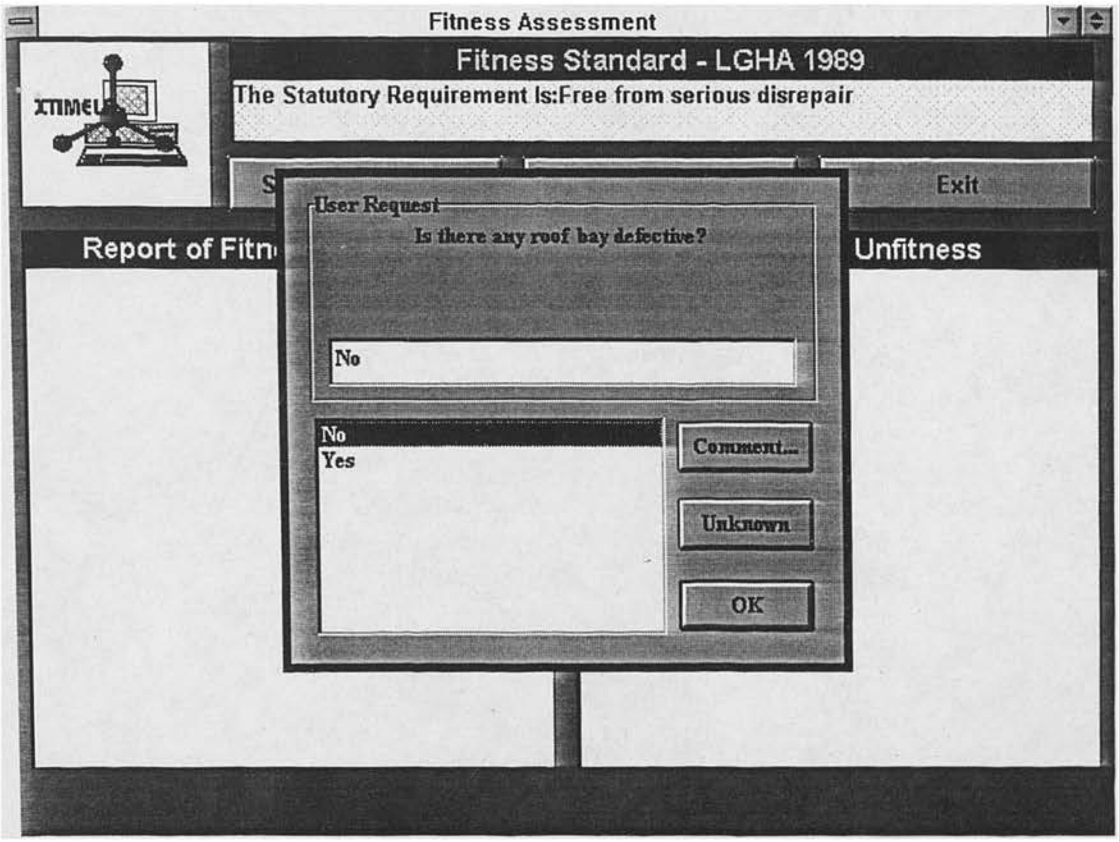
Screen 79: Input information about the condition of the building components



Screen 80: Input information about the condition of the building components

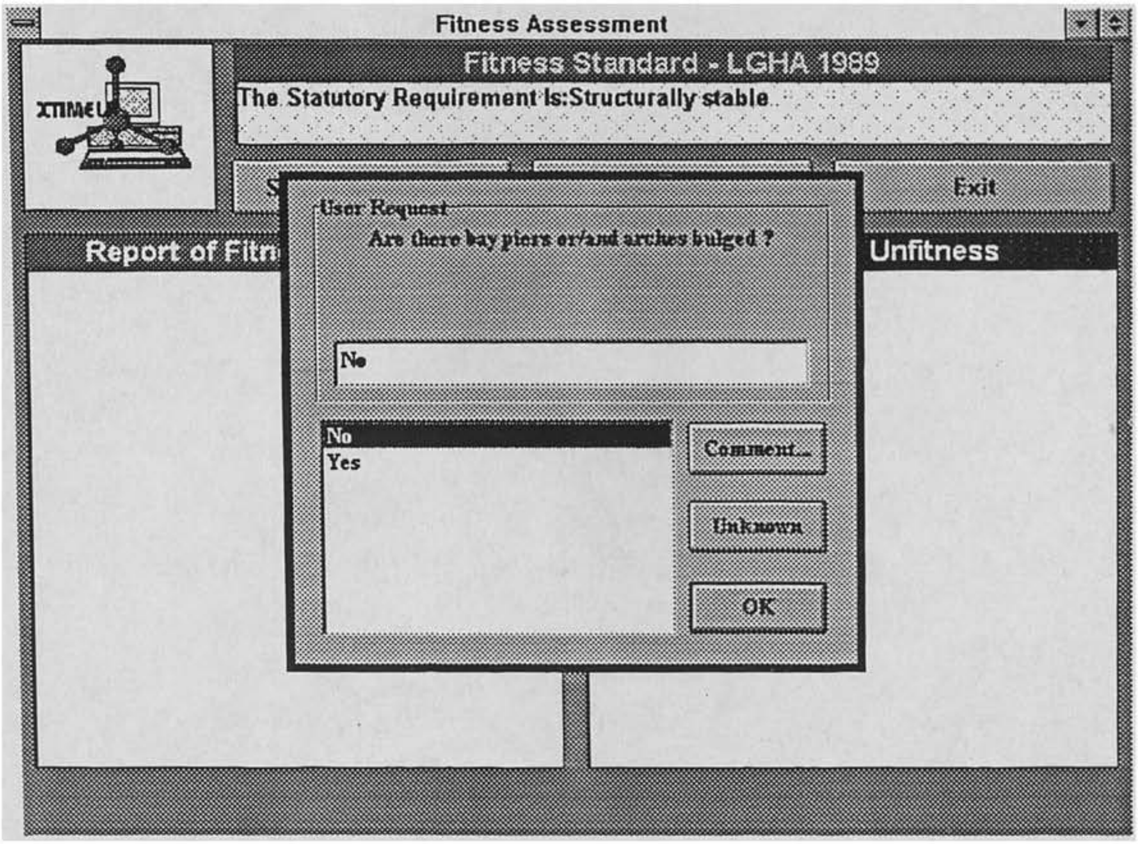


Screen 81: Input information about the condition of the building components

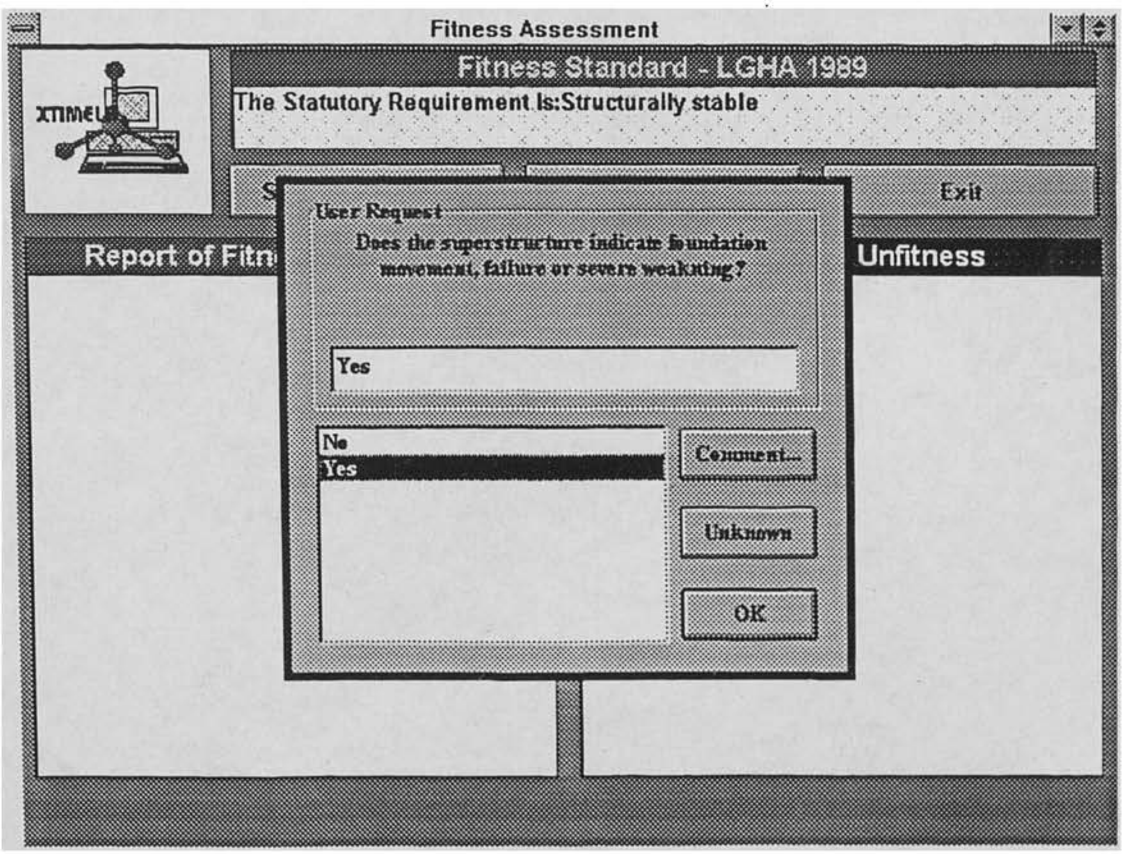


Screen 82: Input information about the condition of the building components

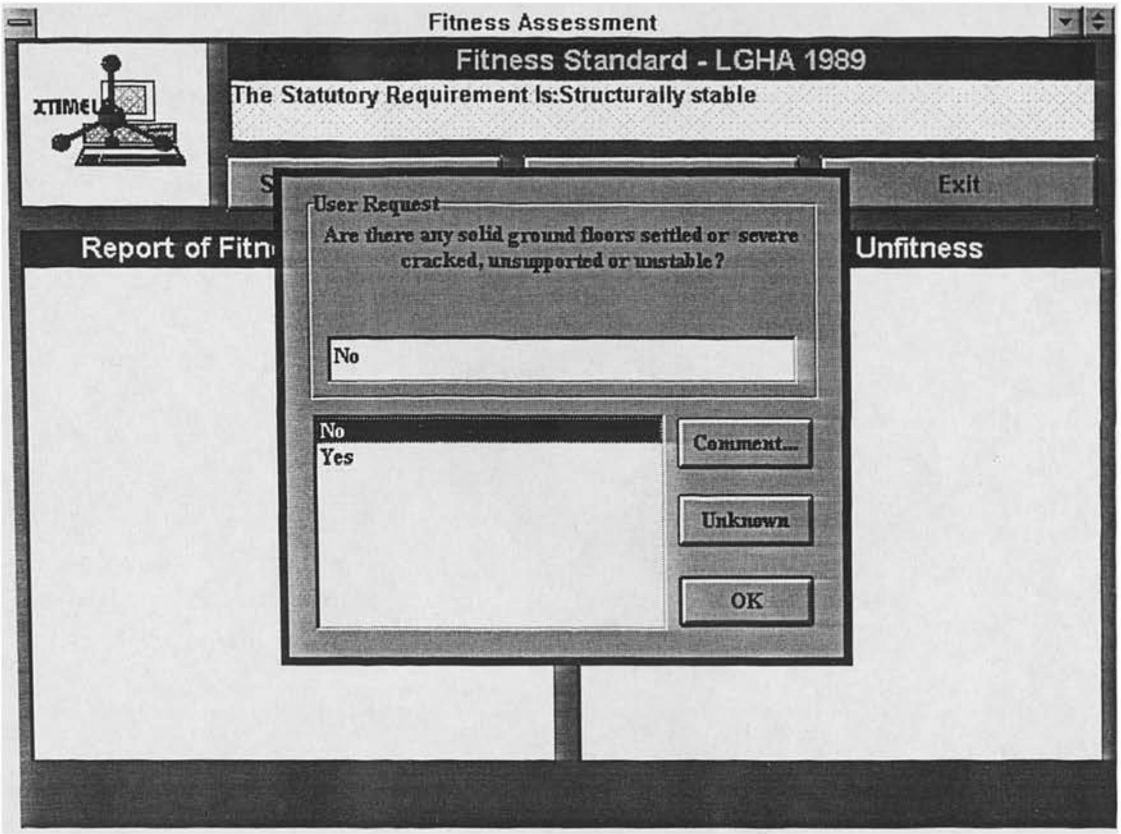
ASSESSING THE STRUCTURAL STABILITY REQUIREMENT



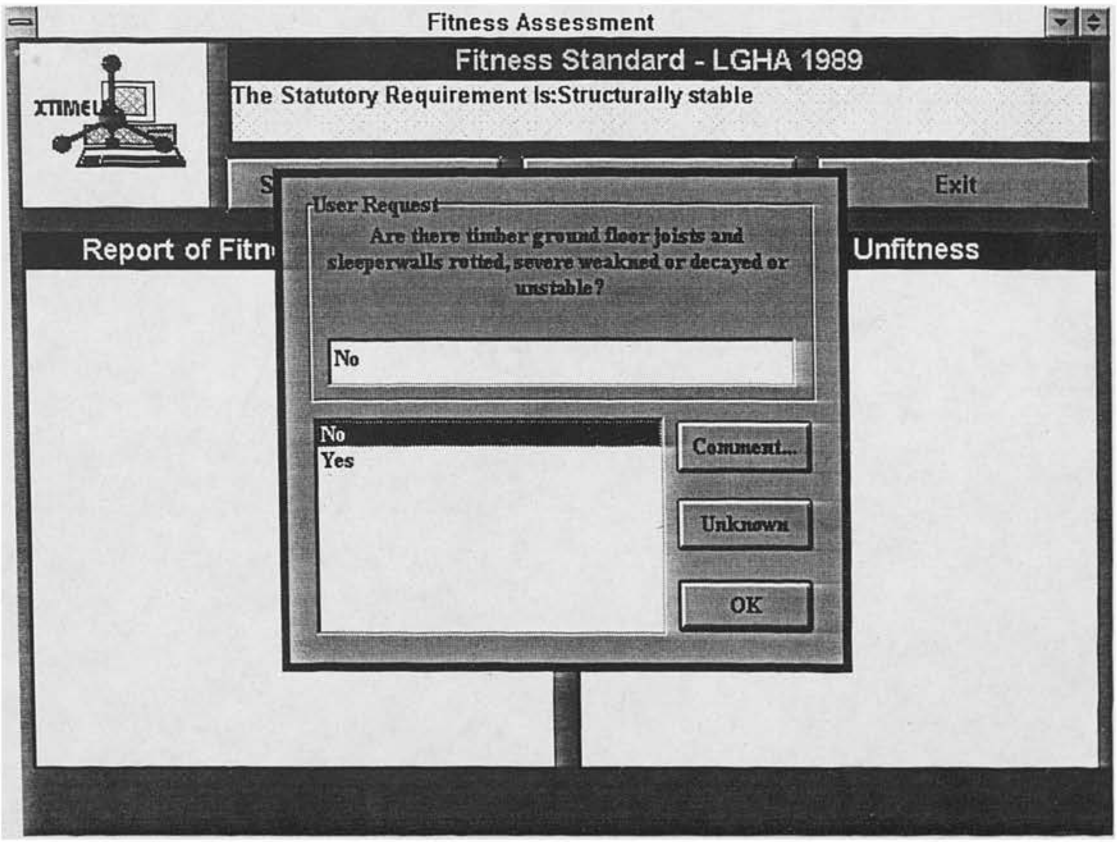
Screen 83: Input information about the condition of the building components



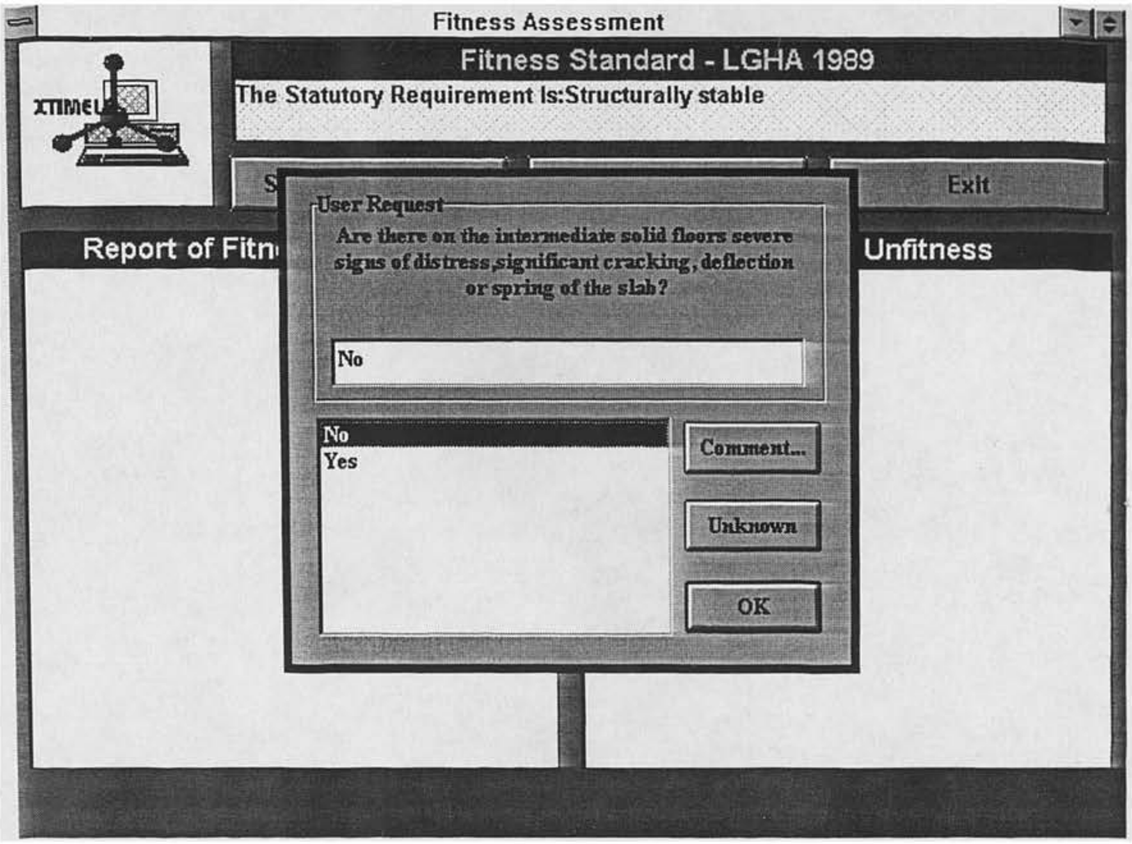
Screen 84: Input information about the condition of the building components



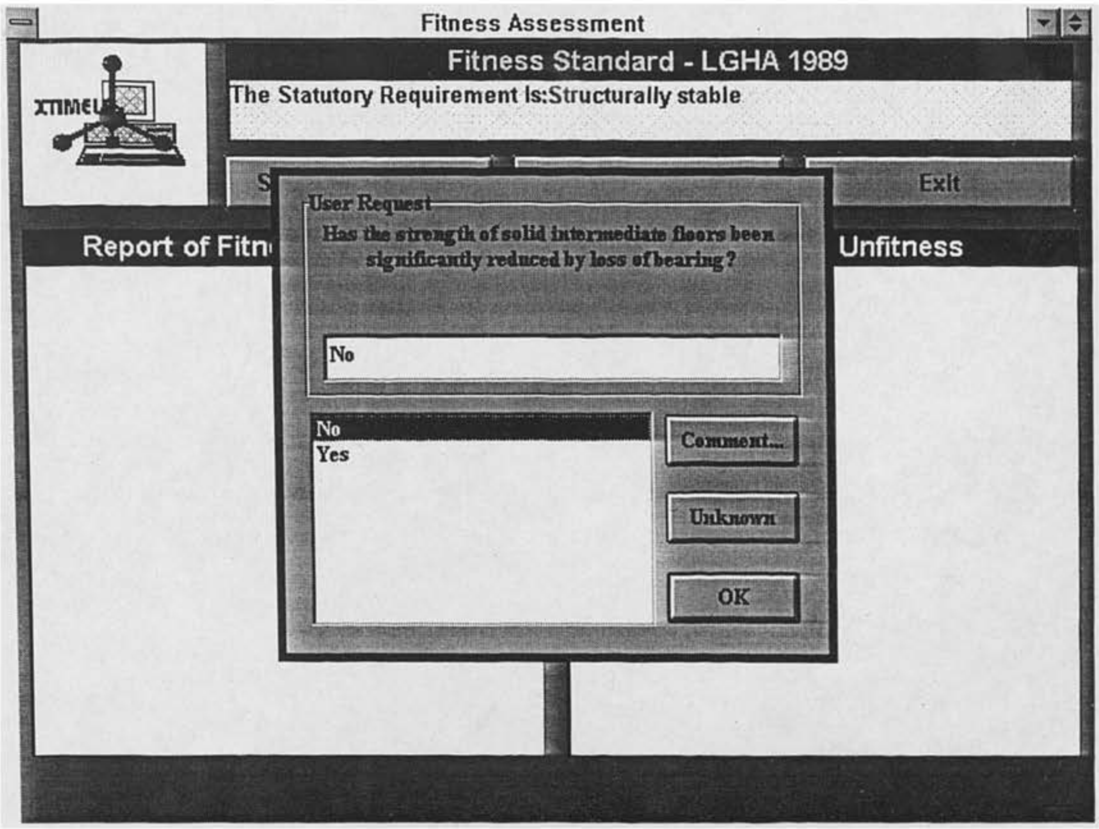
Screen 85: Input information about the condition of the building components



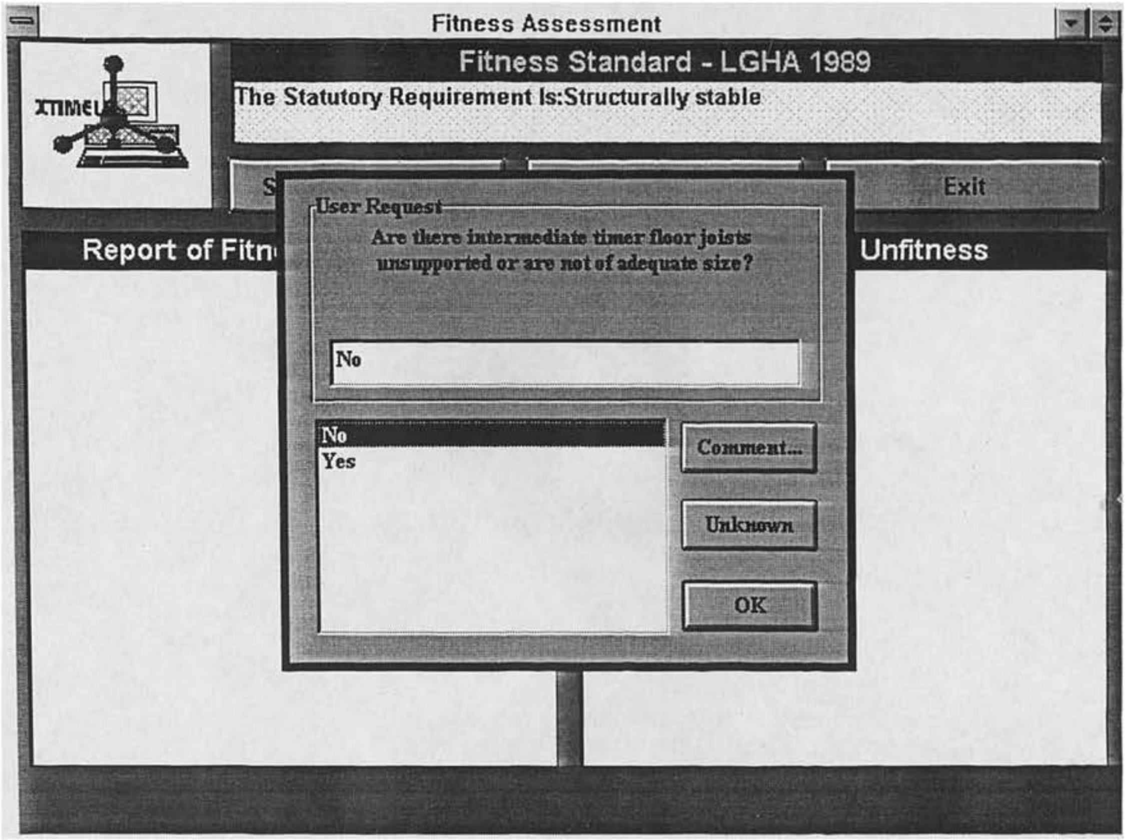
Screen 86: Input information about the condition of the building components



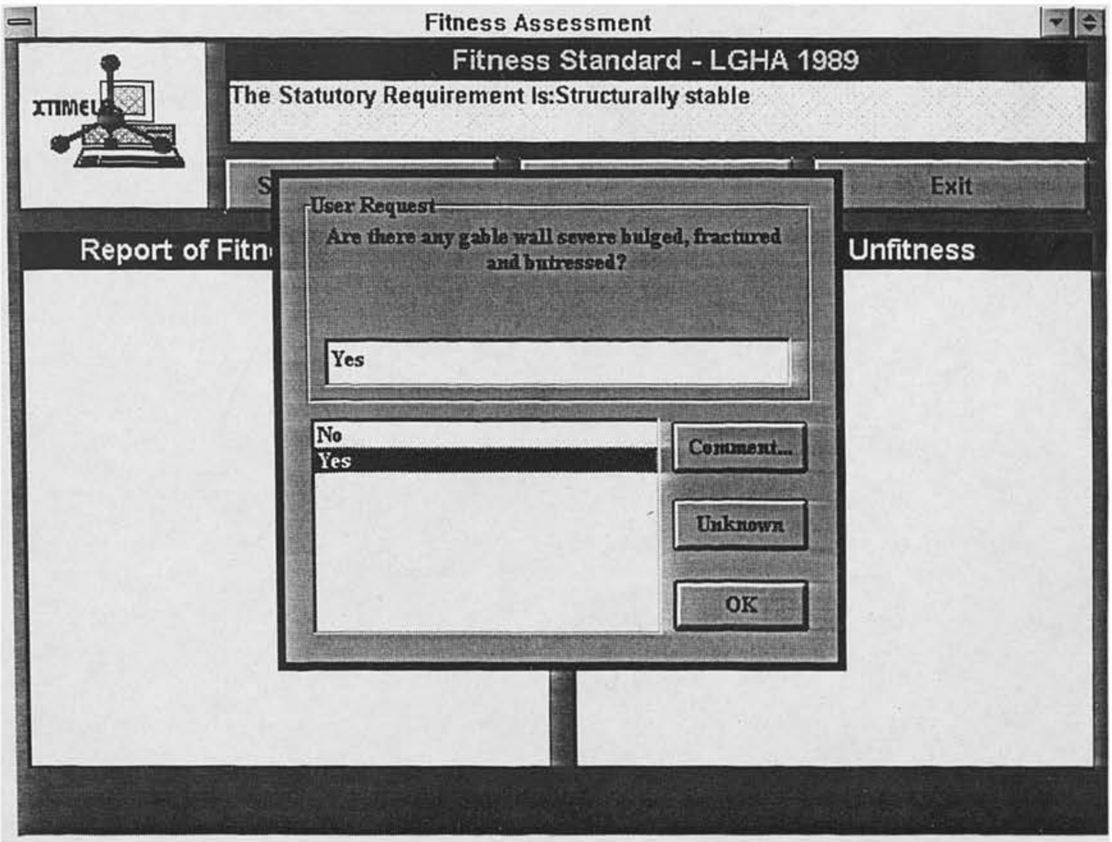
Screen 87: Input information about the condition of the building components



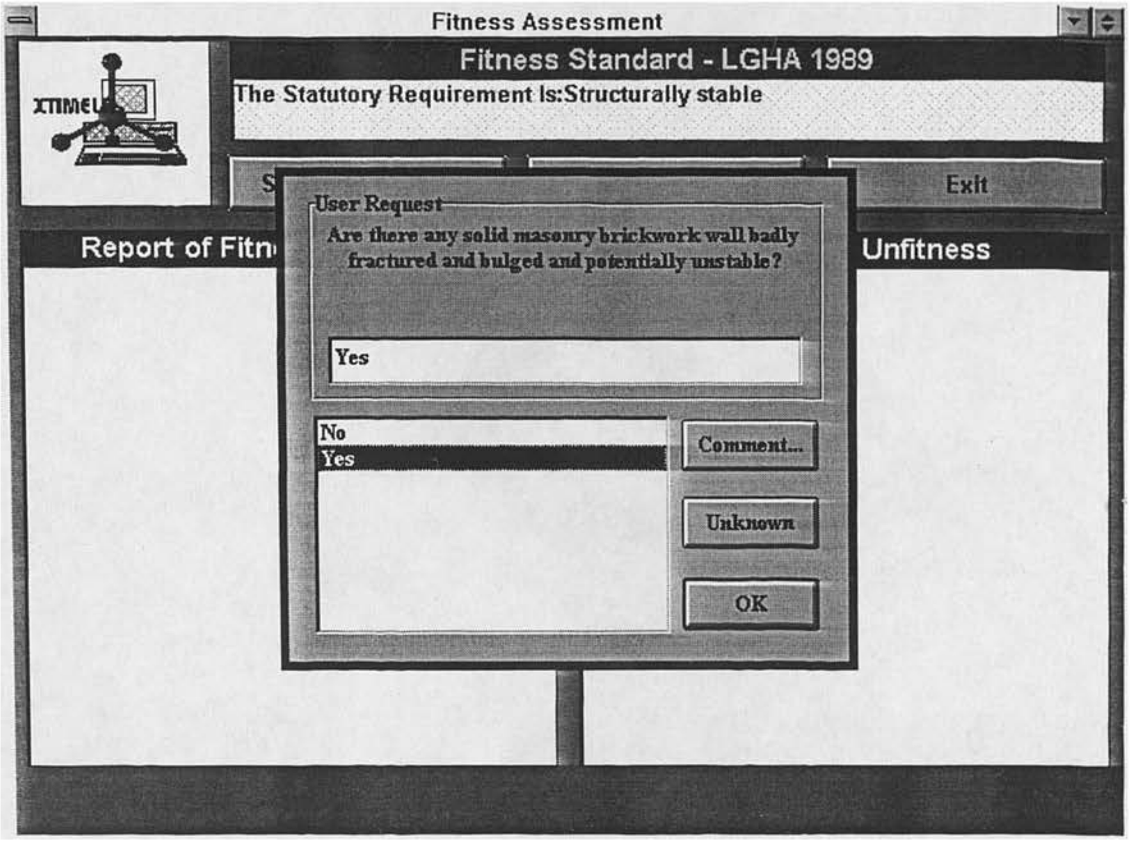
Screen 88: Input information about the condition of the building components



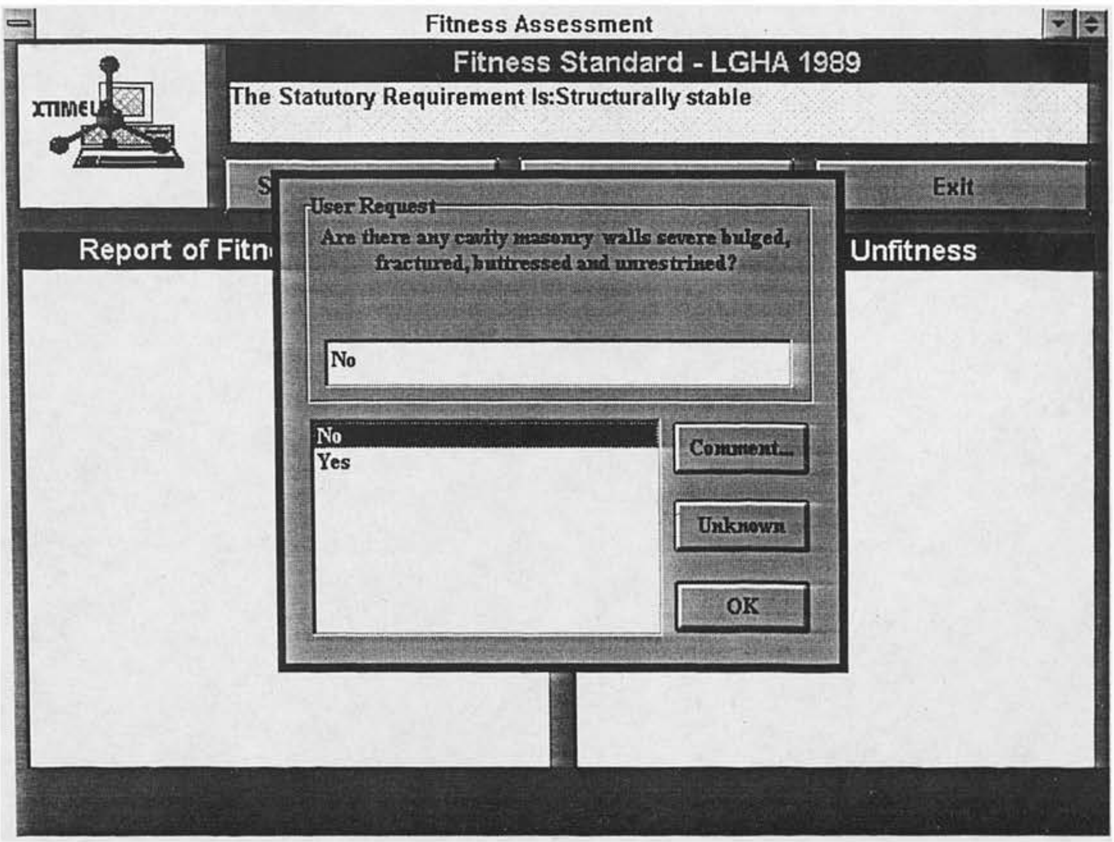
Screen 89: Input information about the condition of the building components



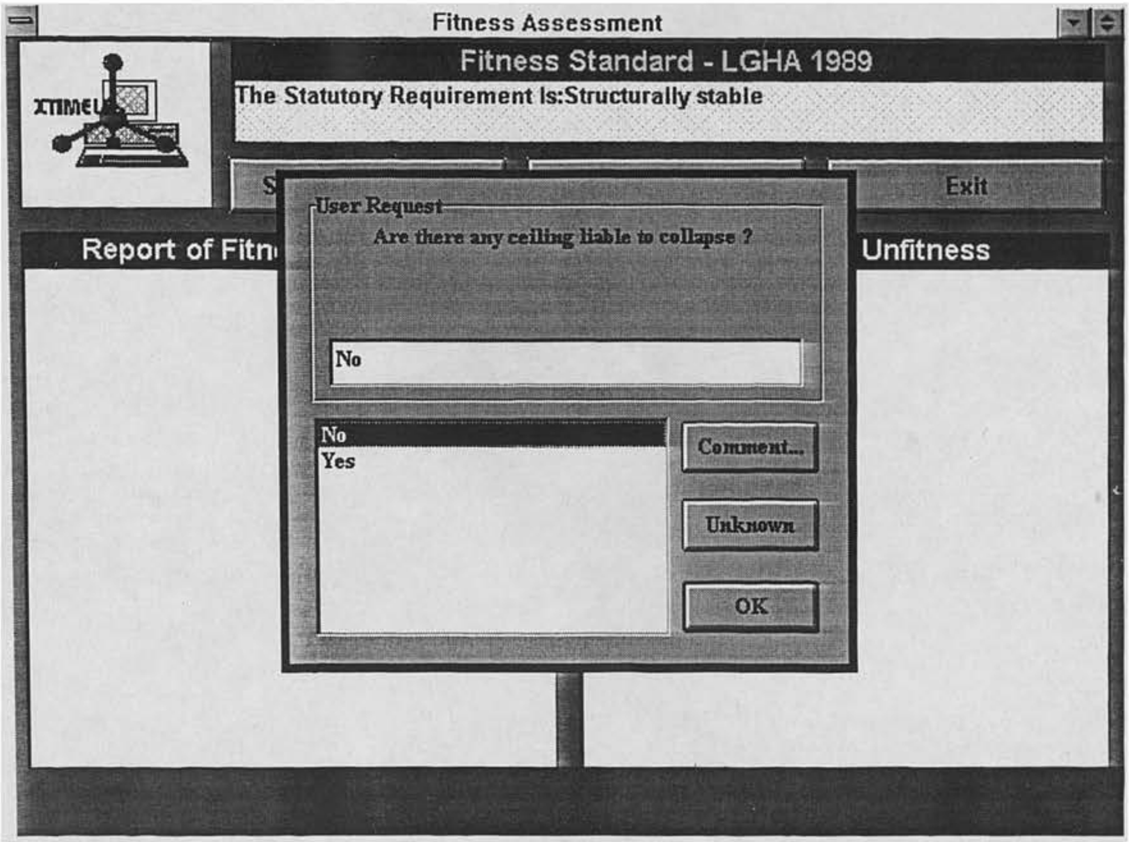
Screen 90: Input information about the condition of the building components



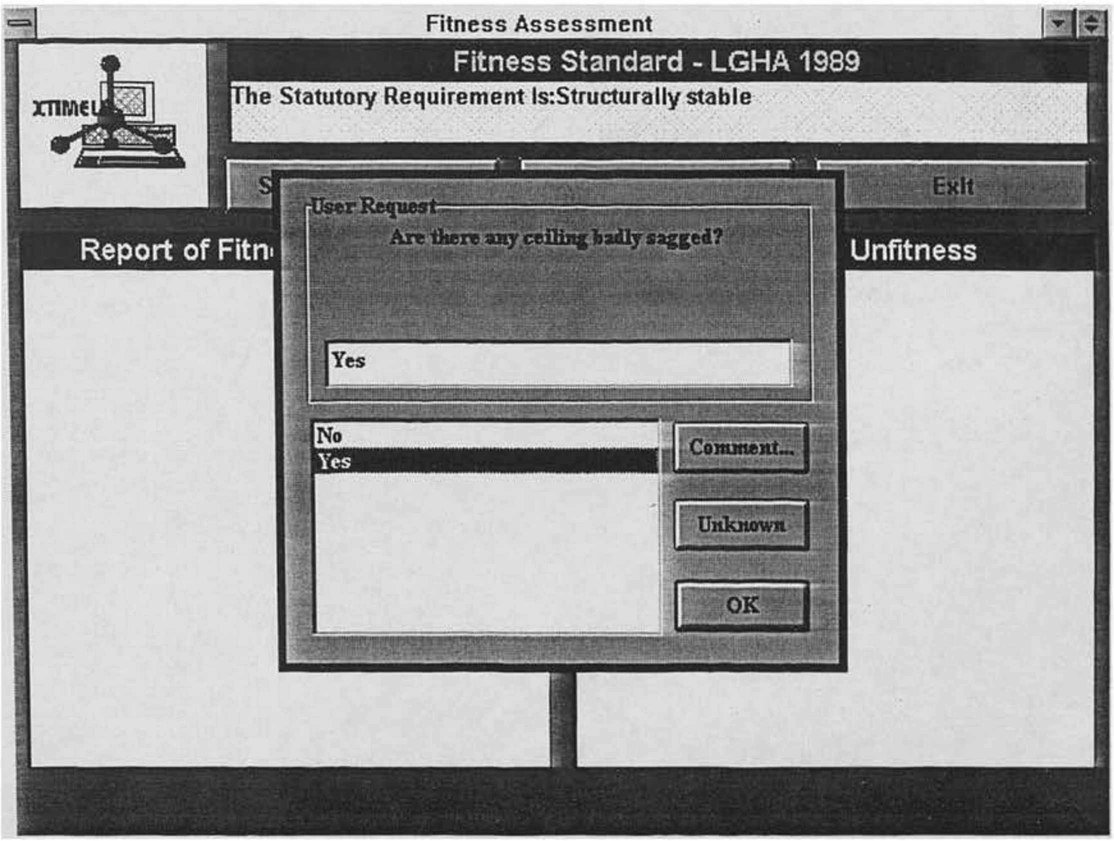
Screen 91: Input information about the condition of the building components



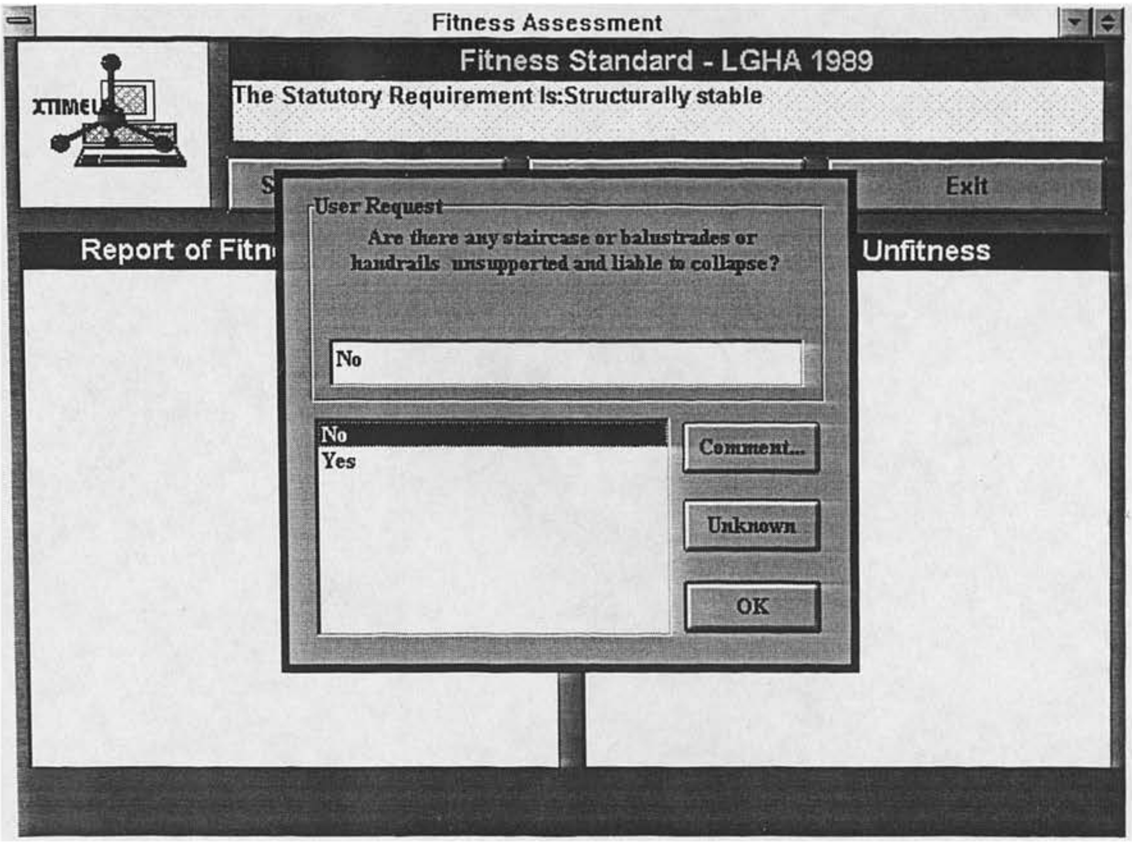
Screen 92: Input information about the condition of the building components



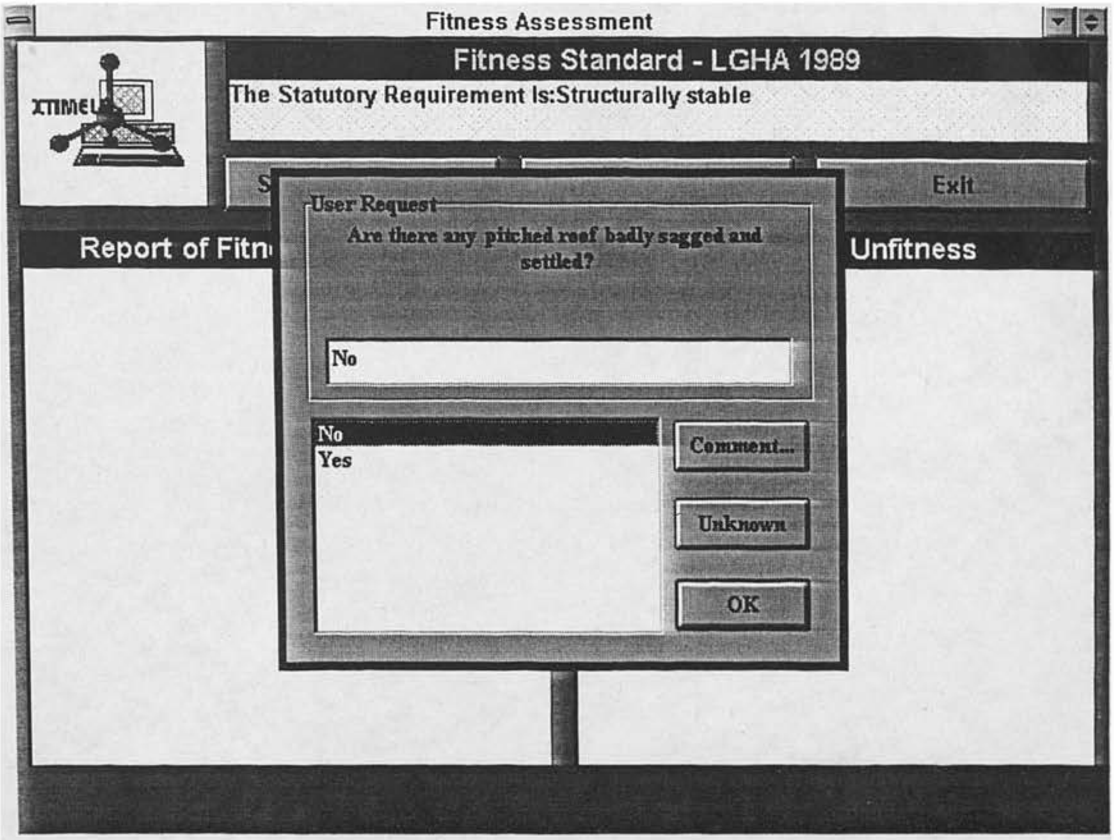
Screen 93: Input information about the condition of the building components



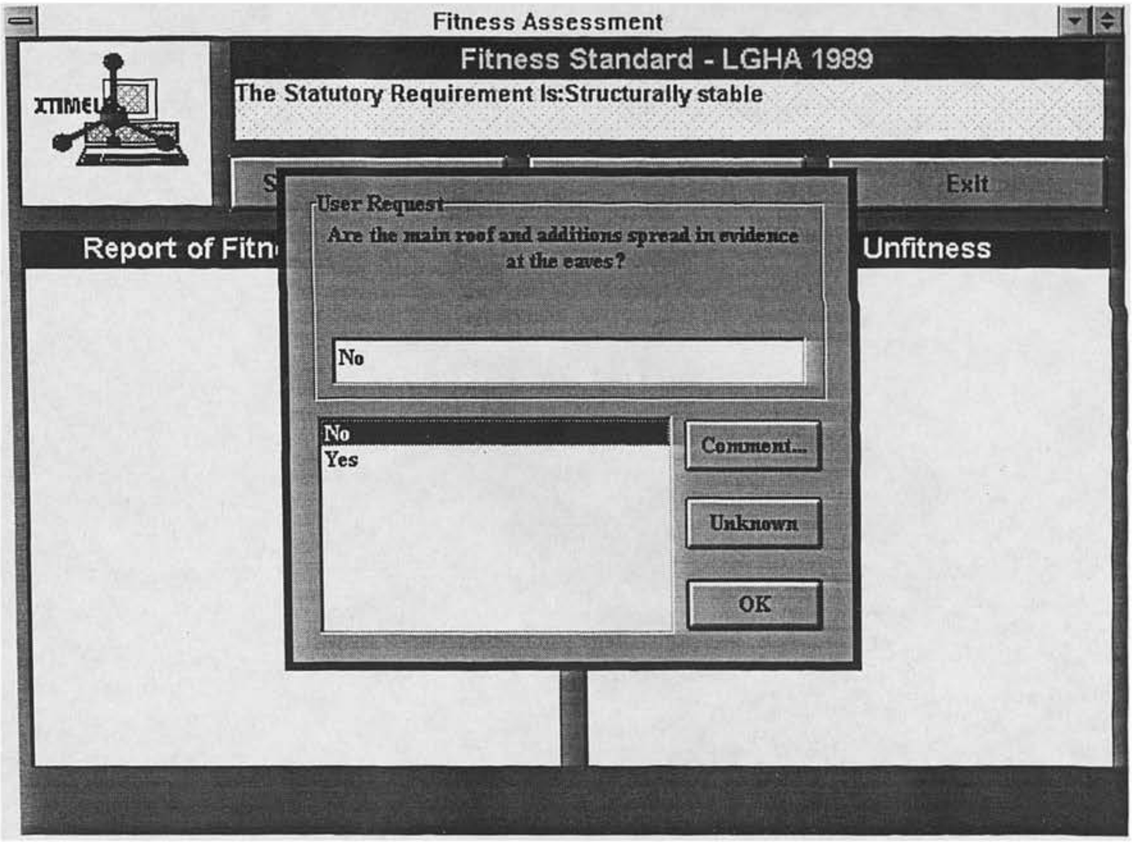
Screen 94: Input information about the condition of the building components



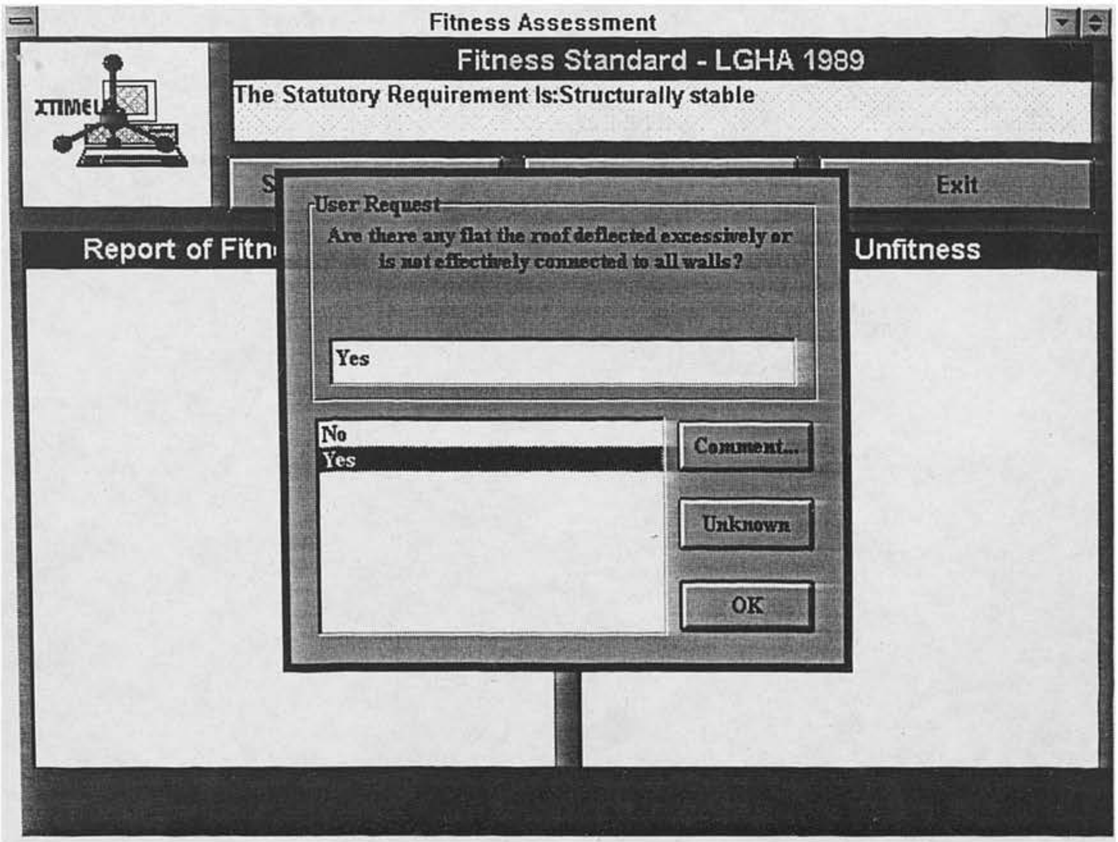
Screen 95: Input information about the condition of the building components



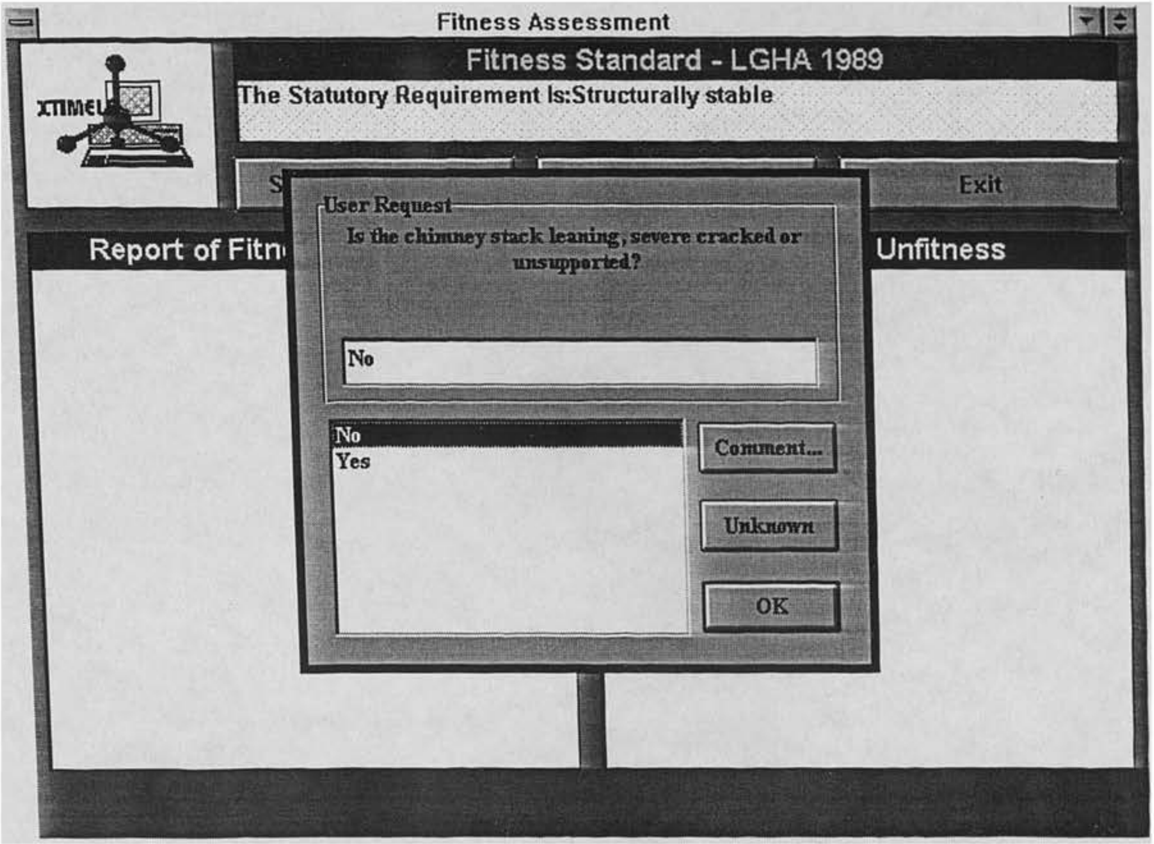
Screen 96: Input information about the condition of the building components



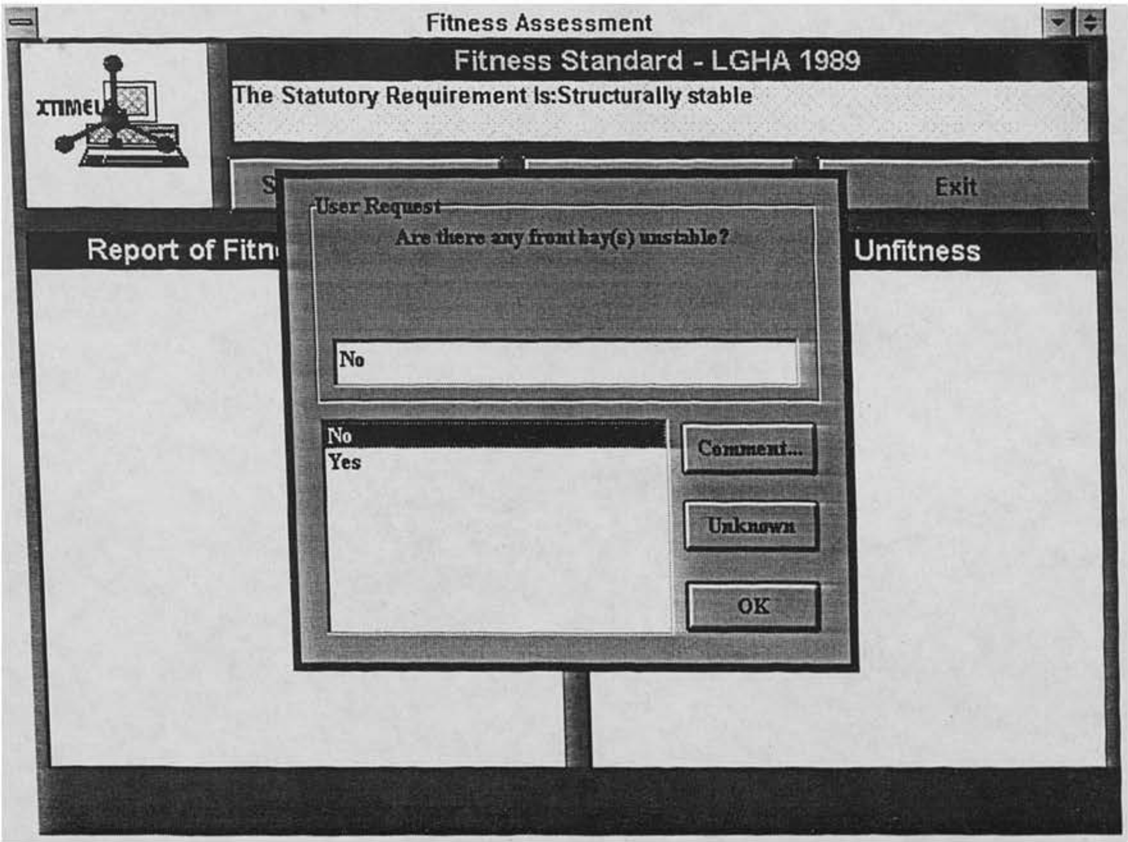
Screen 97: Input information about the condition of the building components



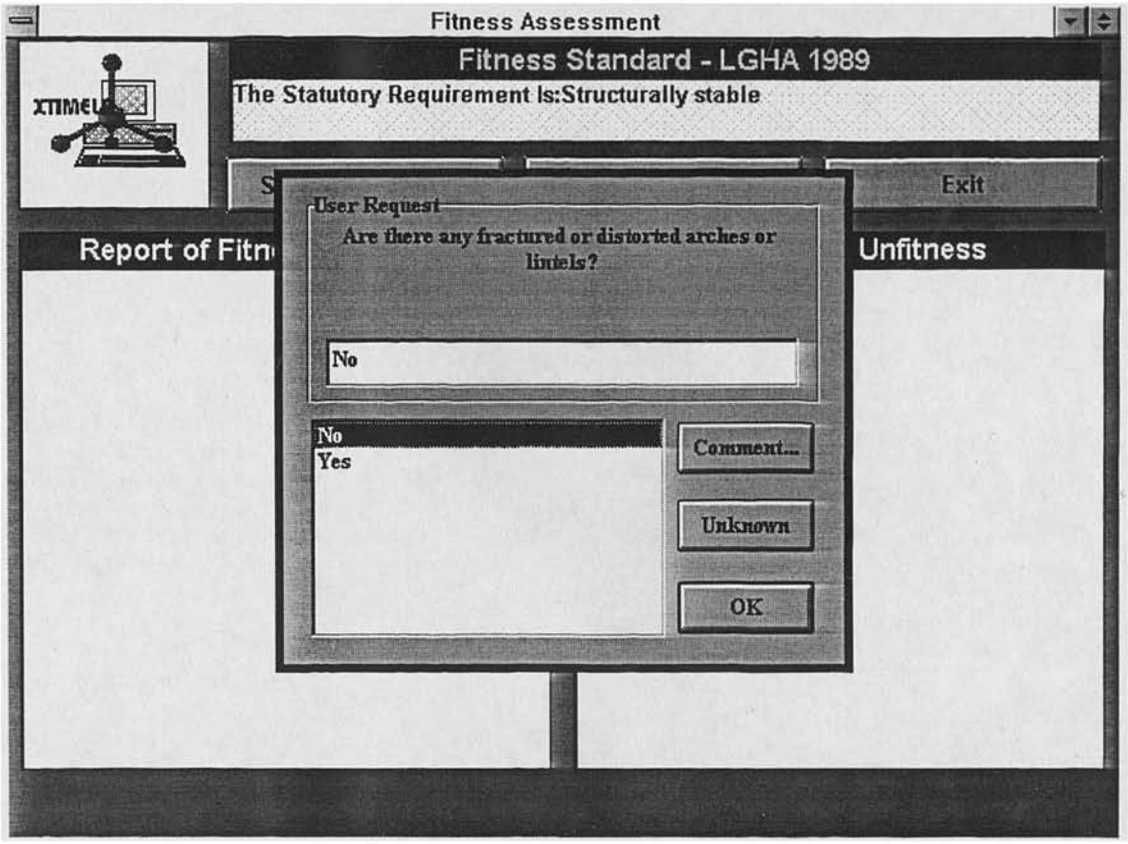
Screen 98: Input information about the condition of the building components



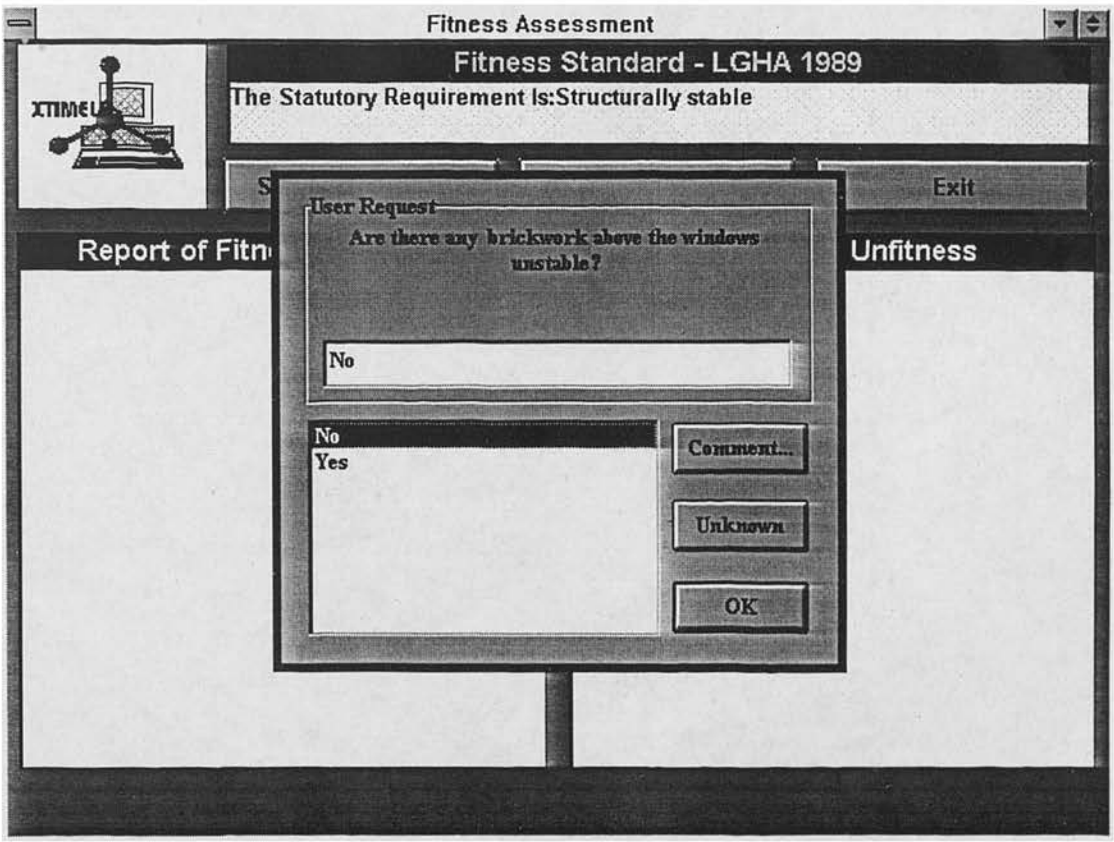
Screen 99: Input information about the condition of the building components



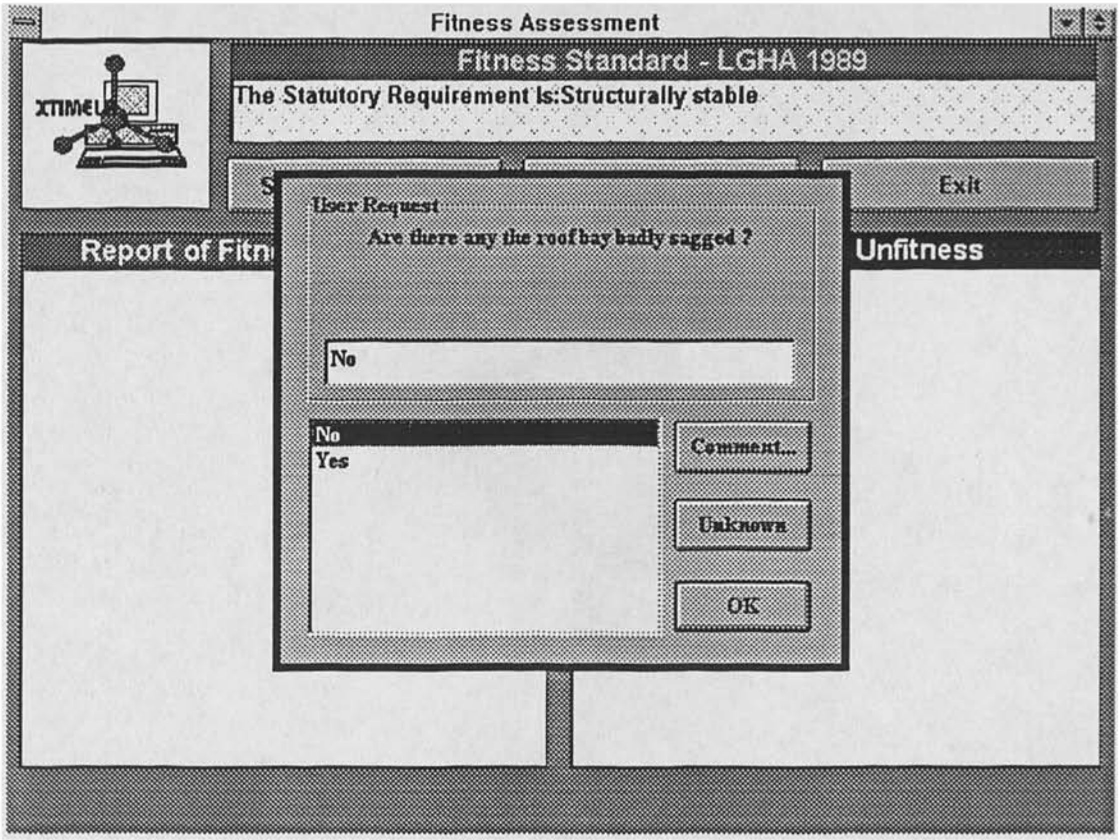
Screen 100: Input information about the condition of the building components



Screen 101: Input information about the condition of the building components

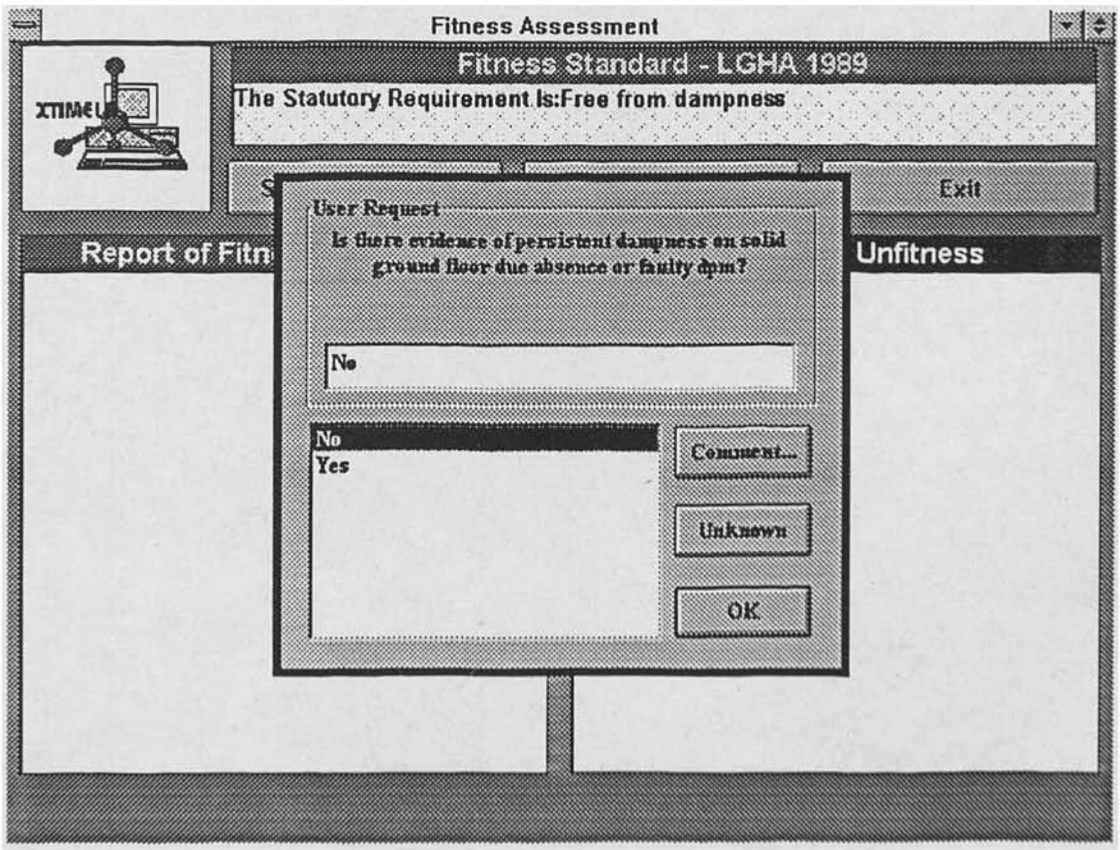


Screen 102: Input information about the condition of the building components

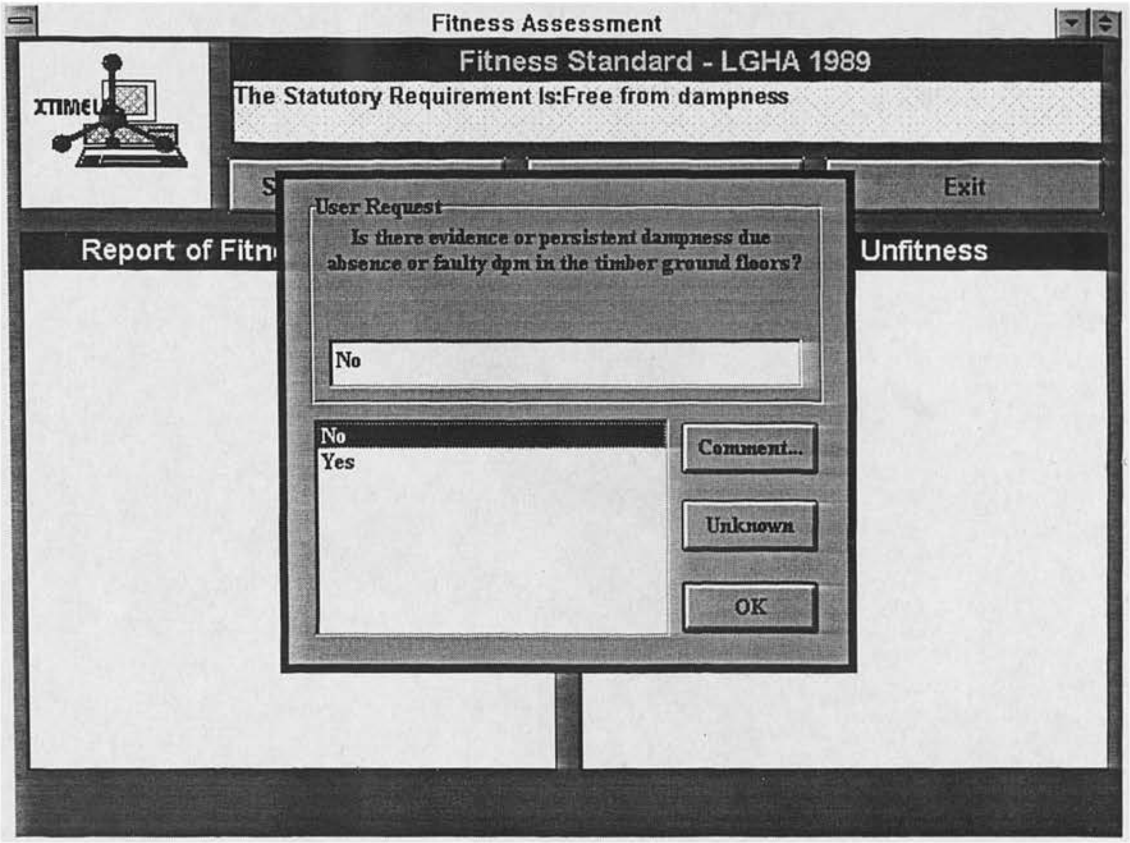


Screen 103: Input information about the condition of the building components

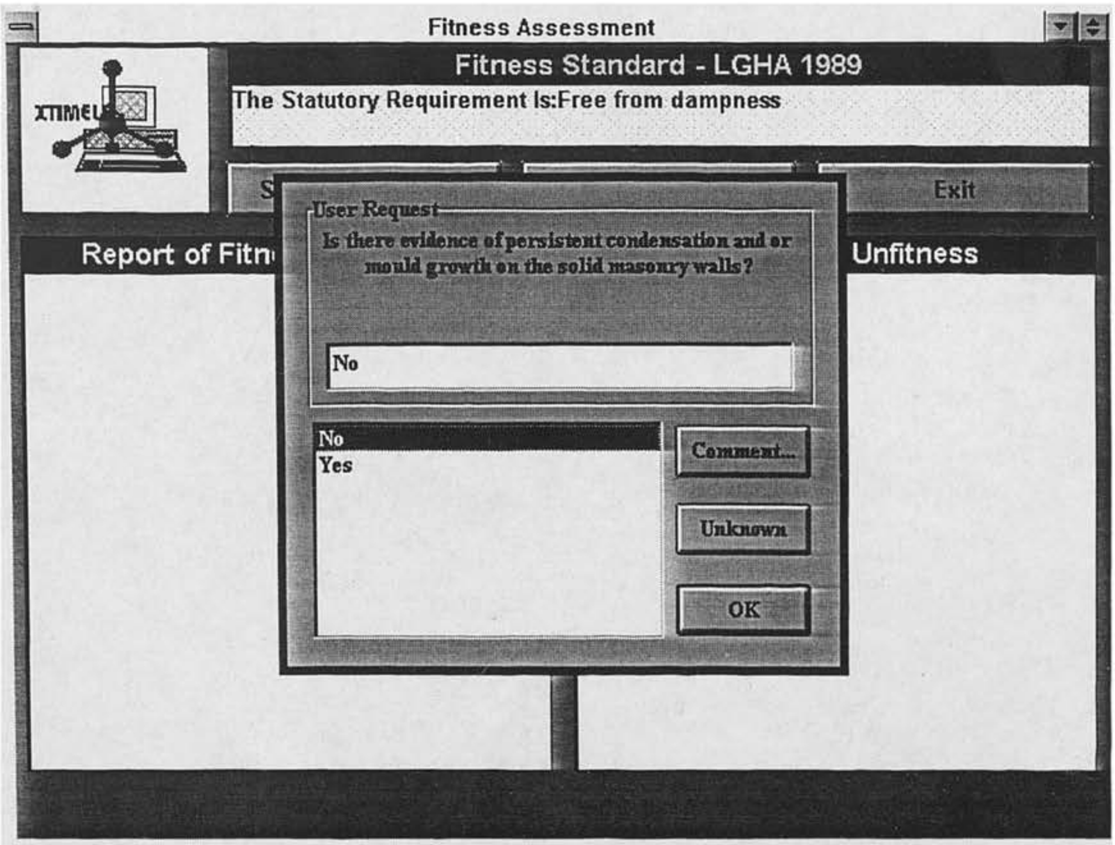
ASSESSING THE DAMPNESS REQUIREMENT



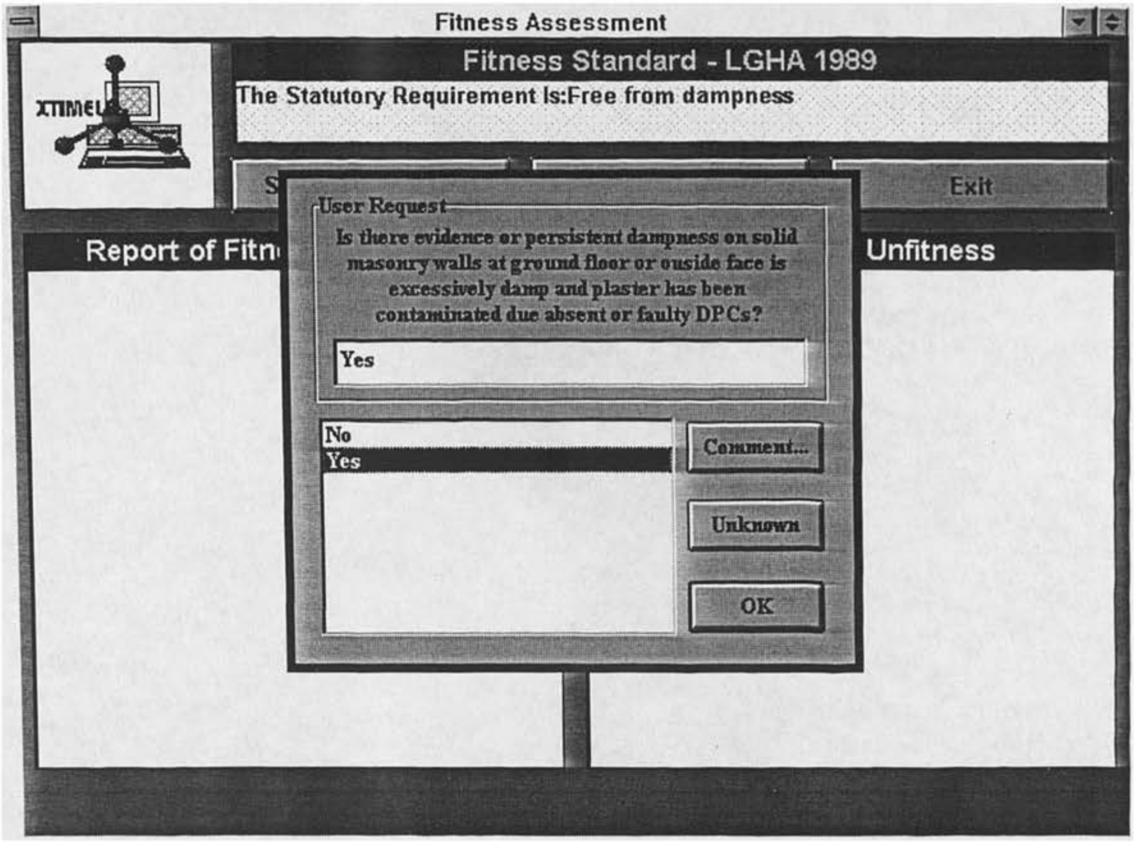
Screen 104: Input information about the condition of the building components



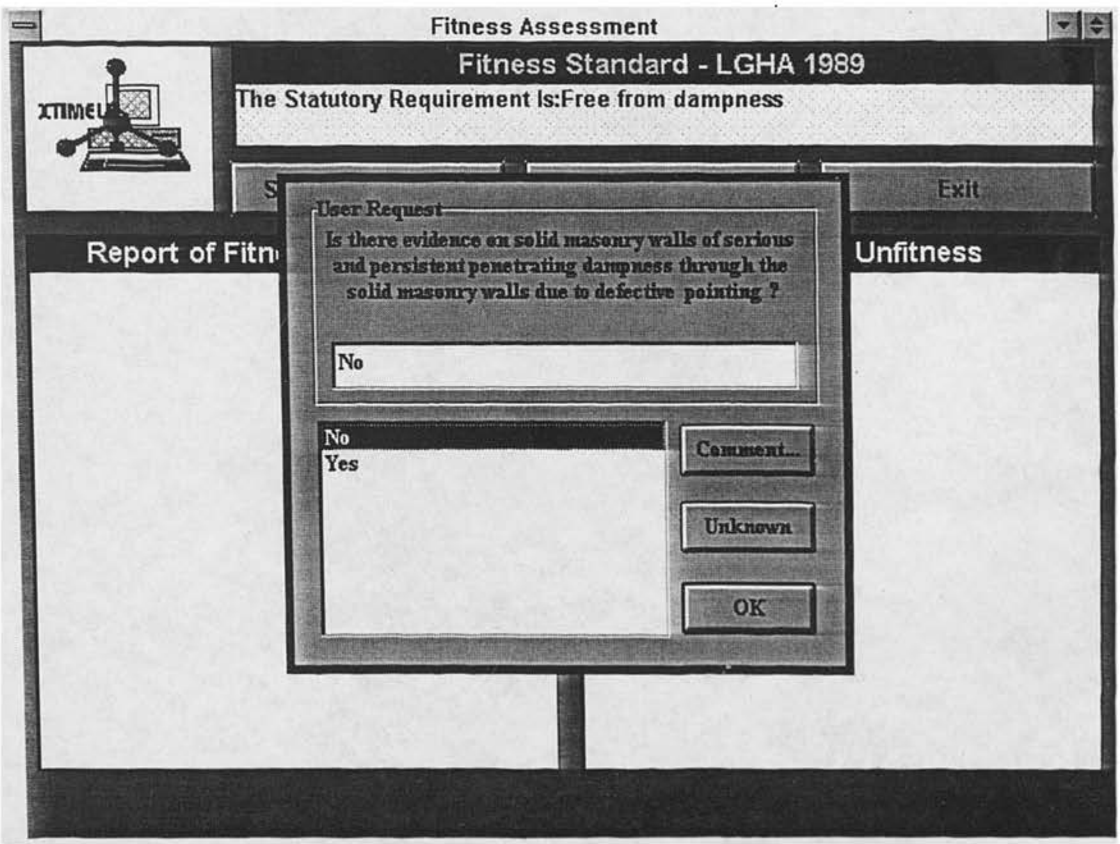
Screen 105: Input information about the condition of the building components



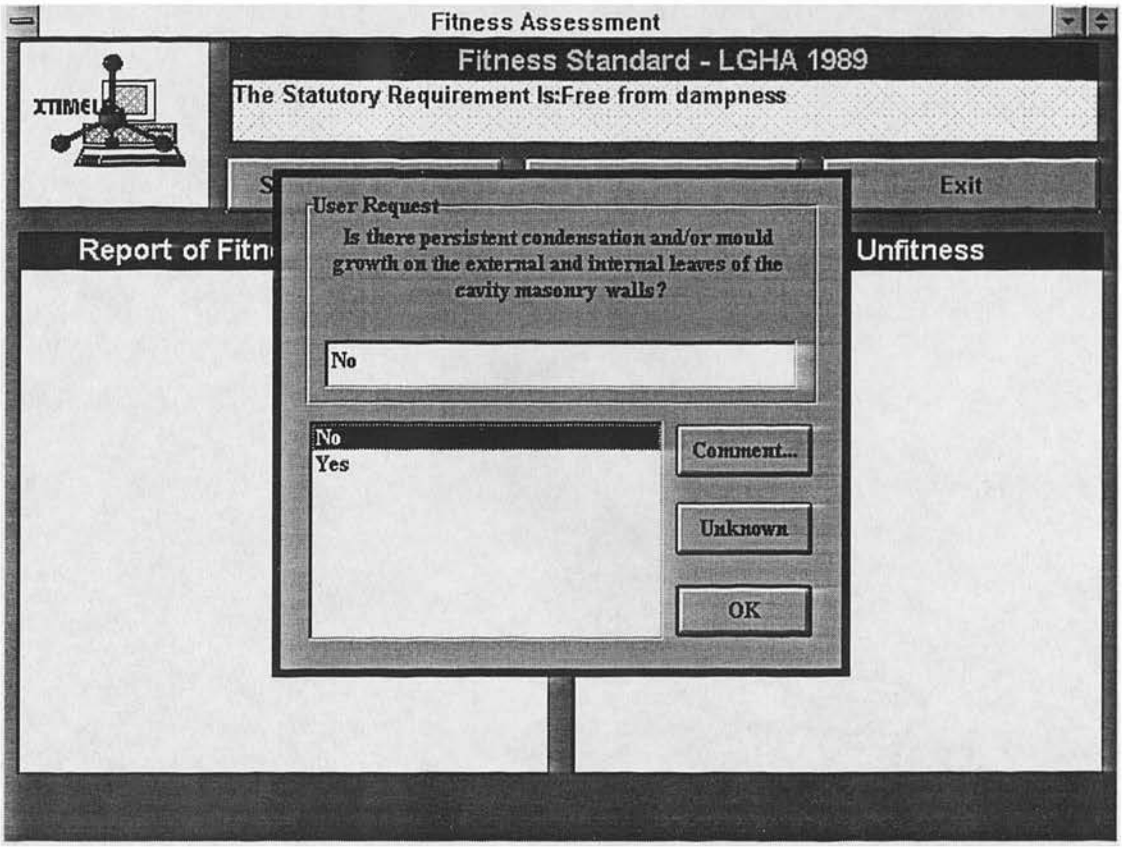
Screen 106: Input information about the condition of the building components



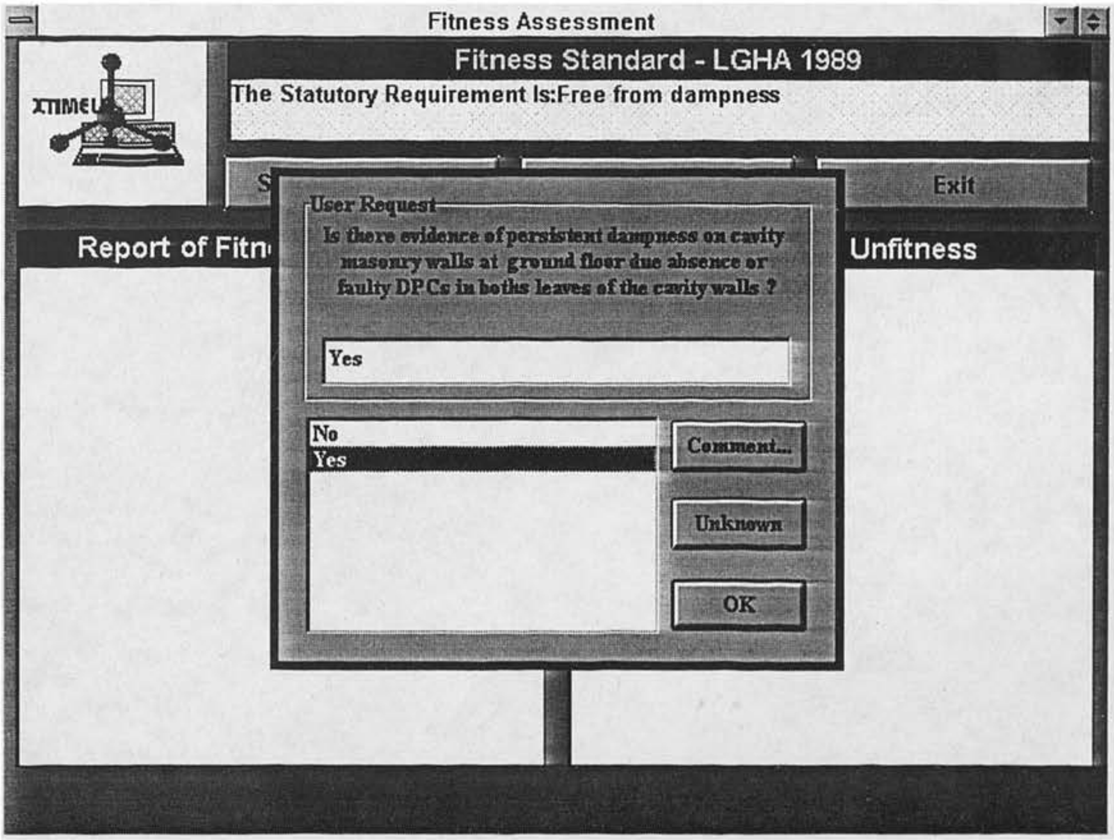
Screen 107: Input information about the condition of the building components



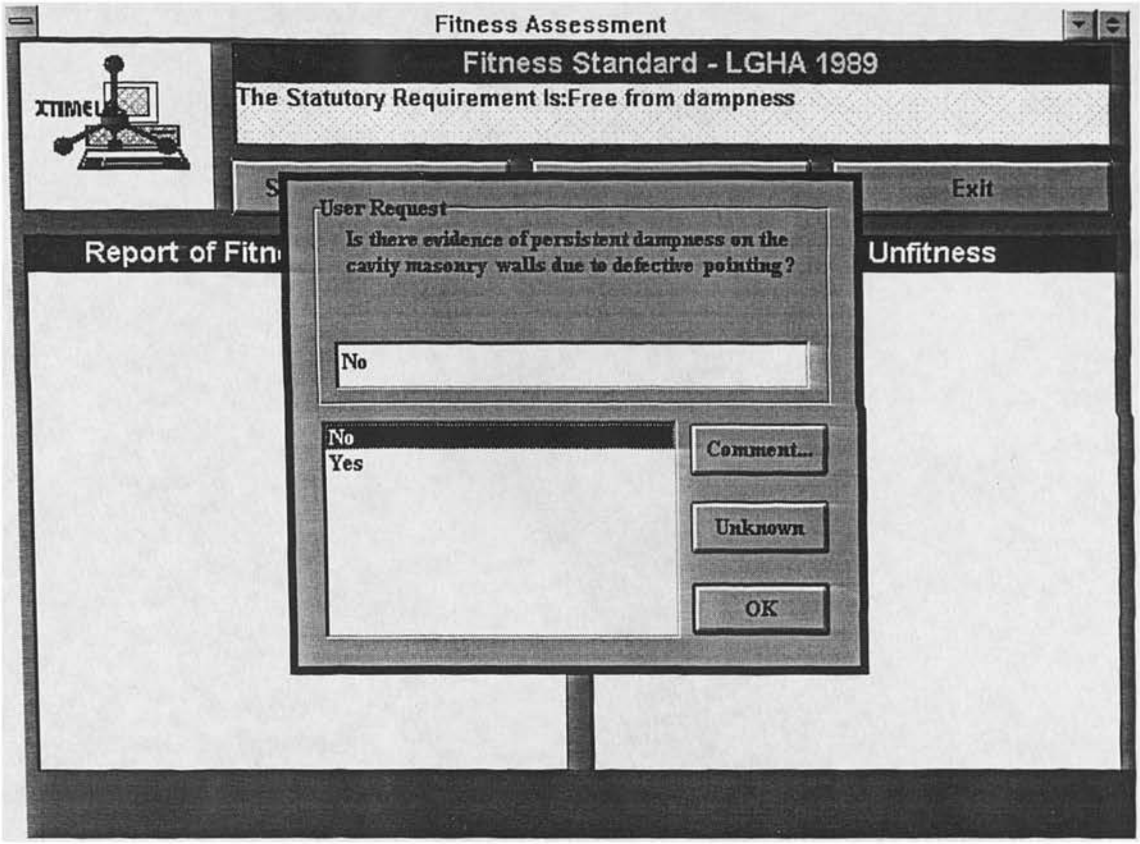
Screen 108: Input information about the condition of the building components



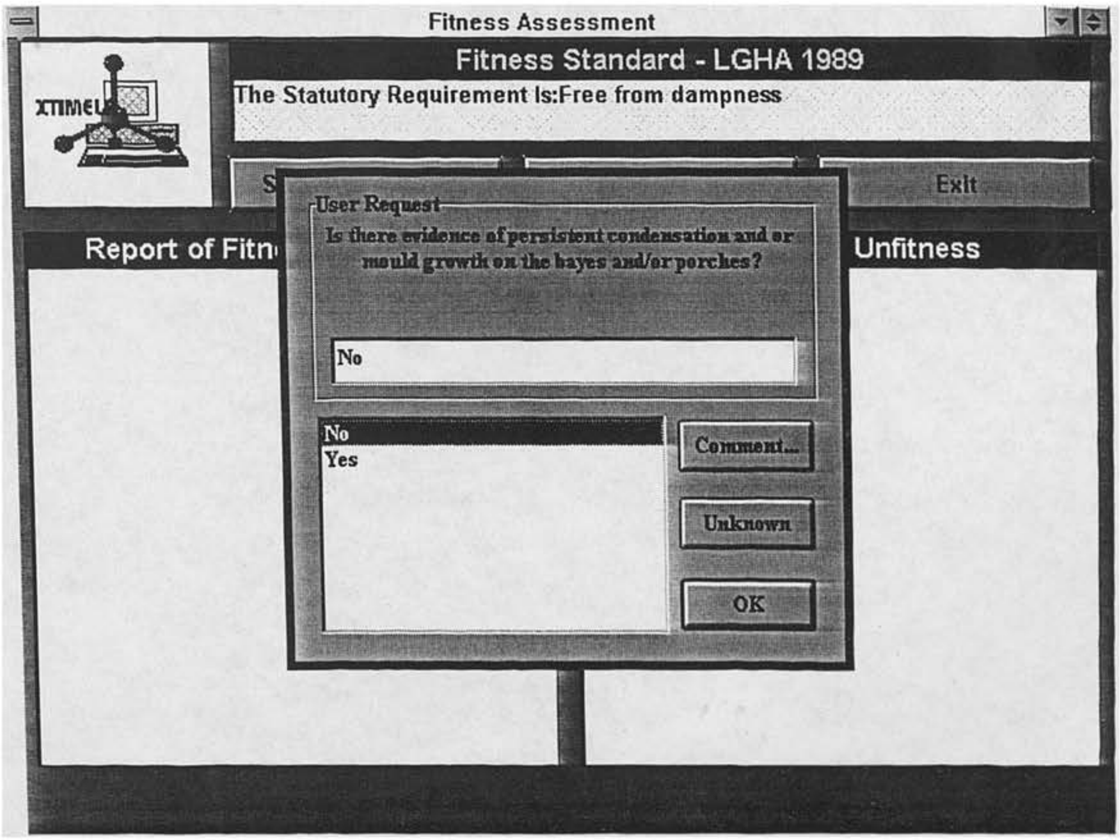
Screen 109: Input information about the condition of the building components



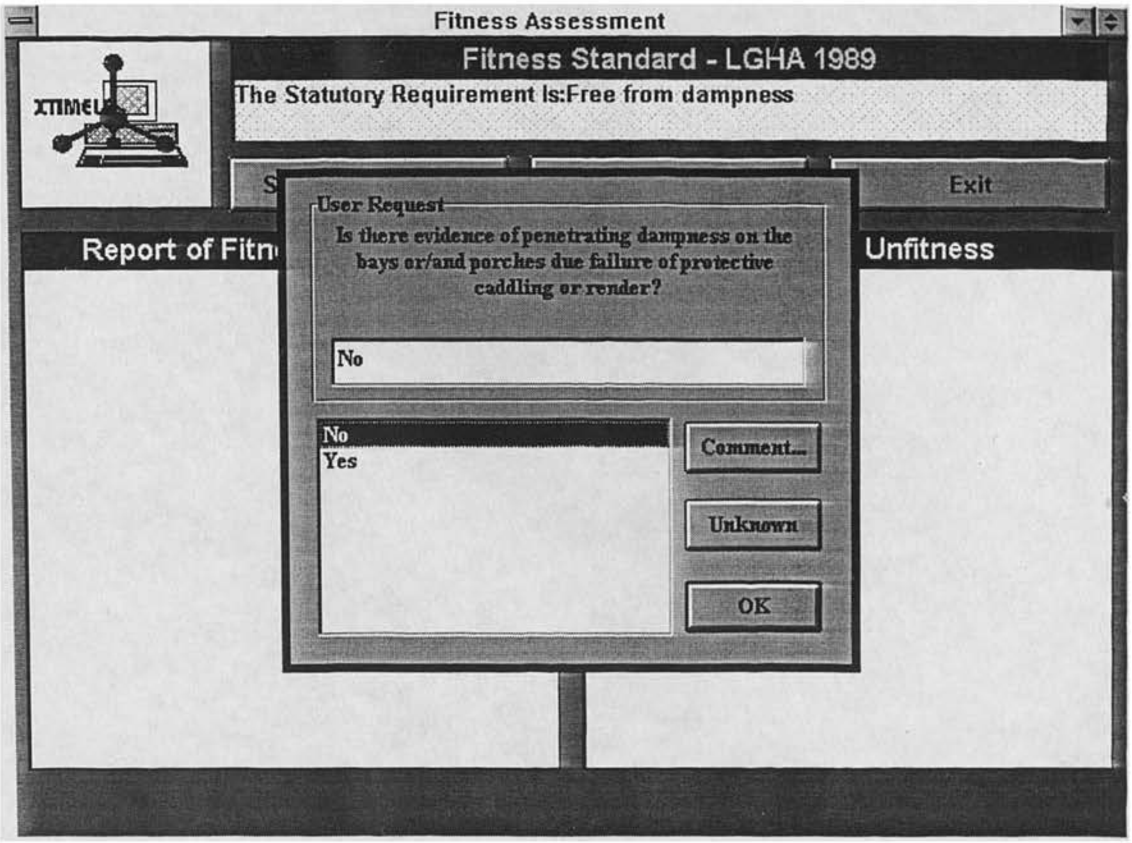
Screen 110: Input information about the condition of the building components



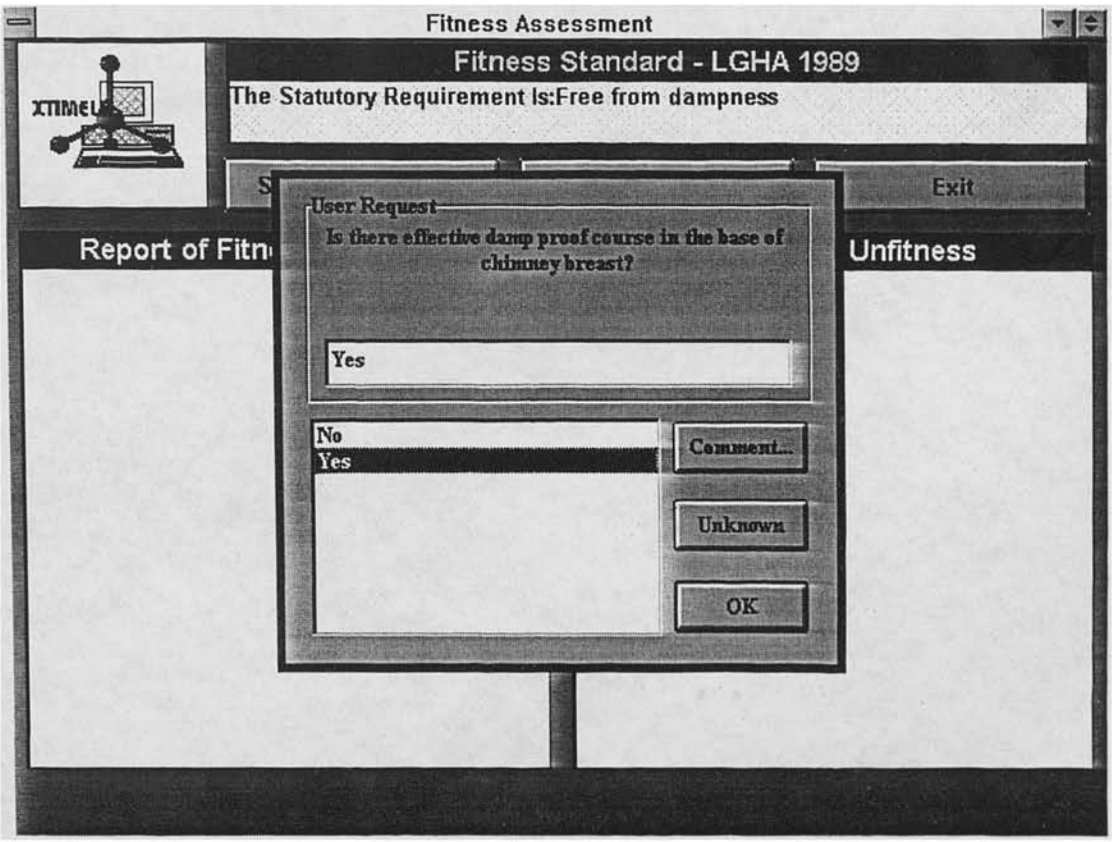
Screen 111: Input information about the condition of the building components



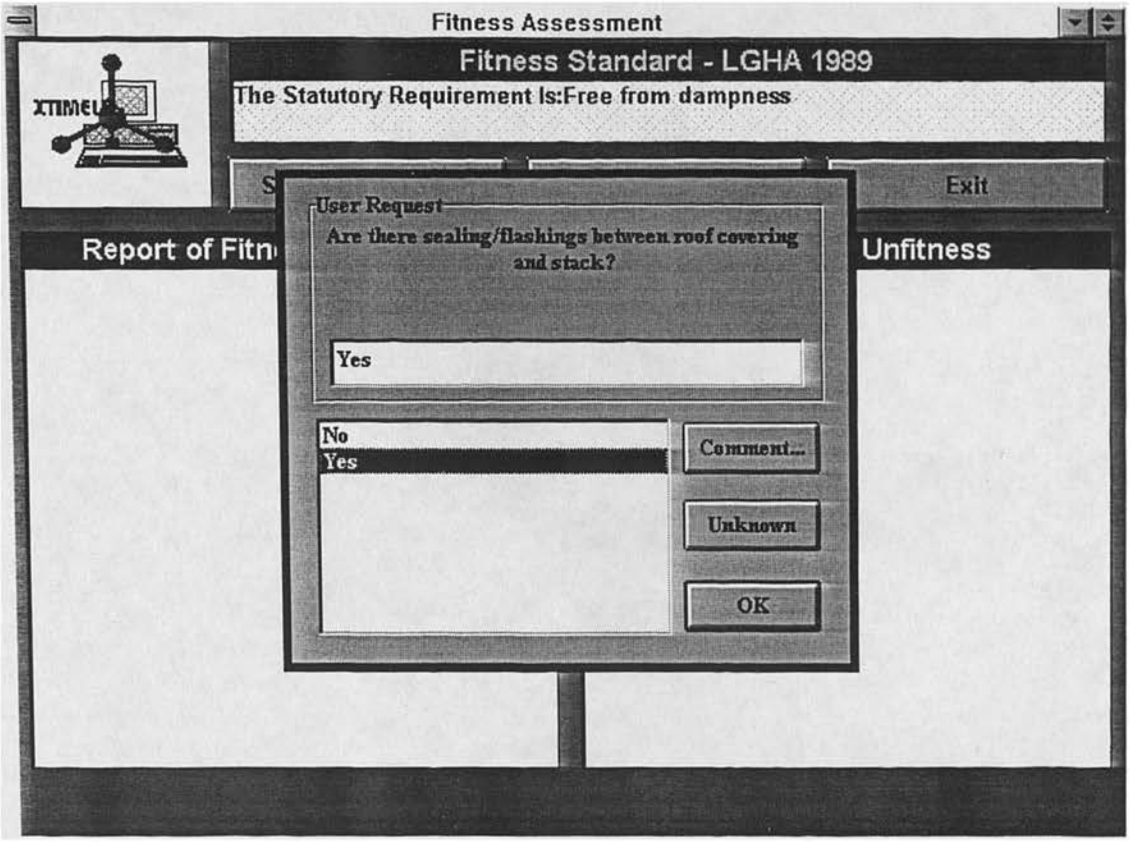
Screen 112: Input information about the condition of the building components



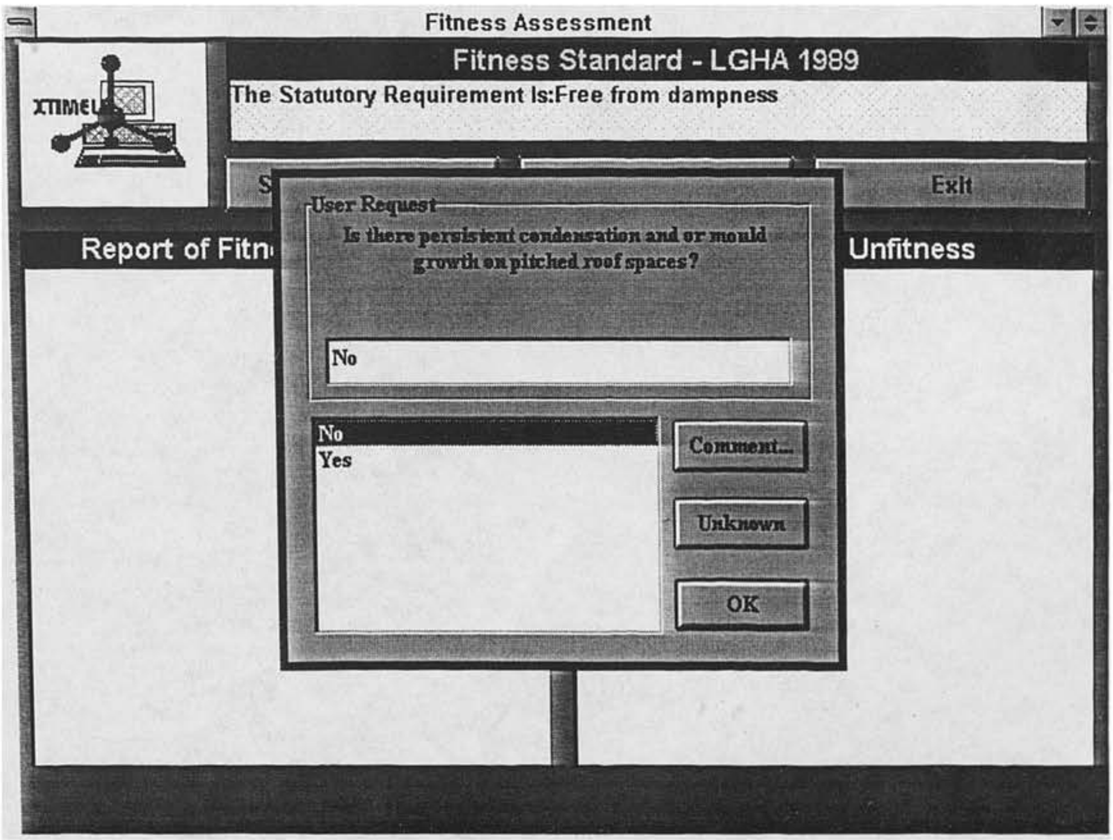
Screen 113: Input information about the condition of the building components



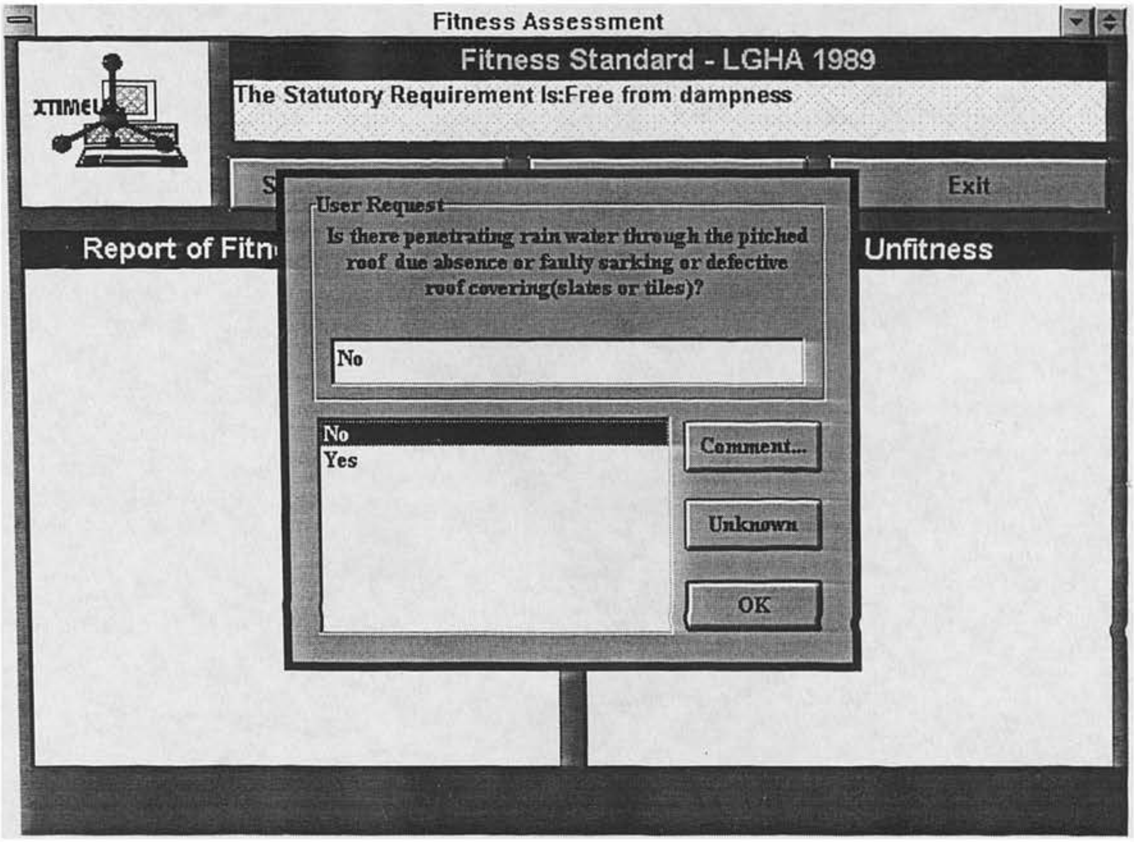
Screen 114: Input information about the condition of the building components



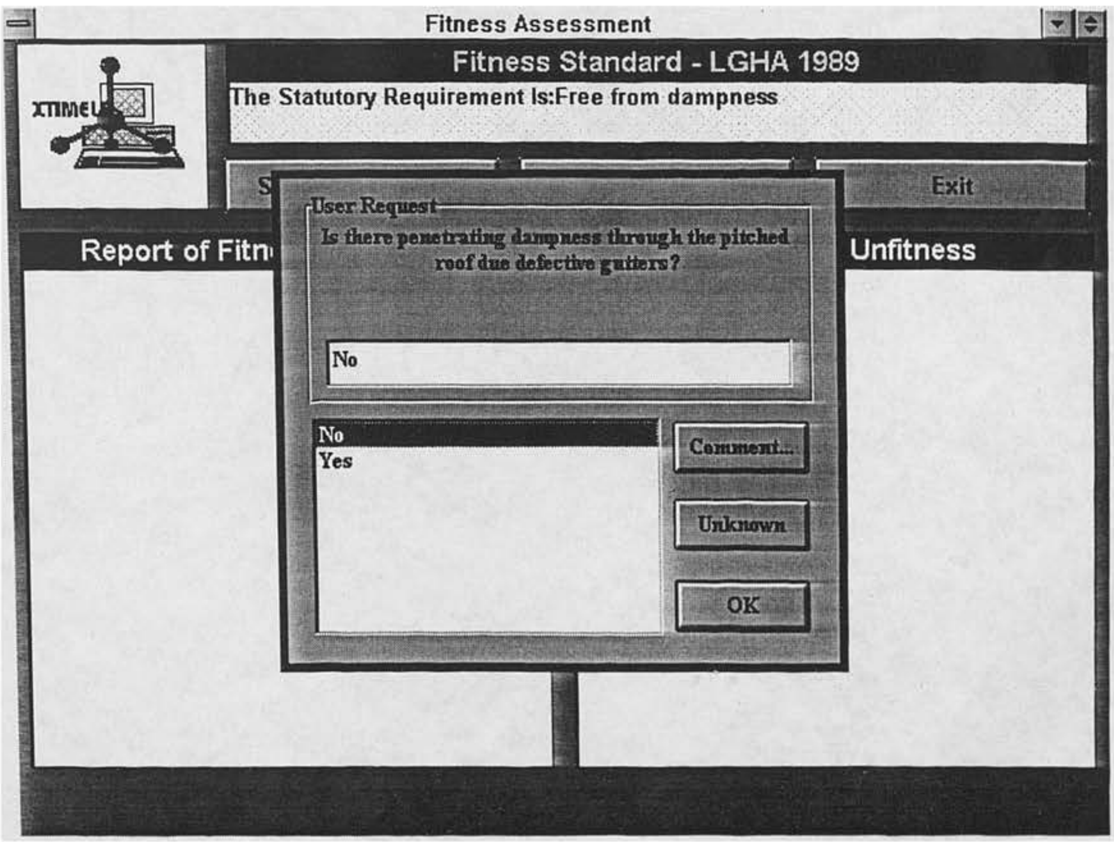
Screen 115: Input information about the condition of the building components



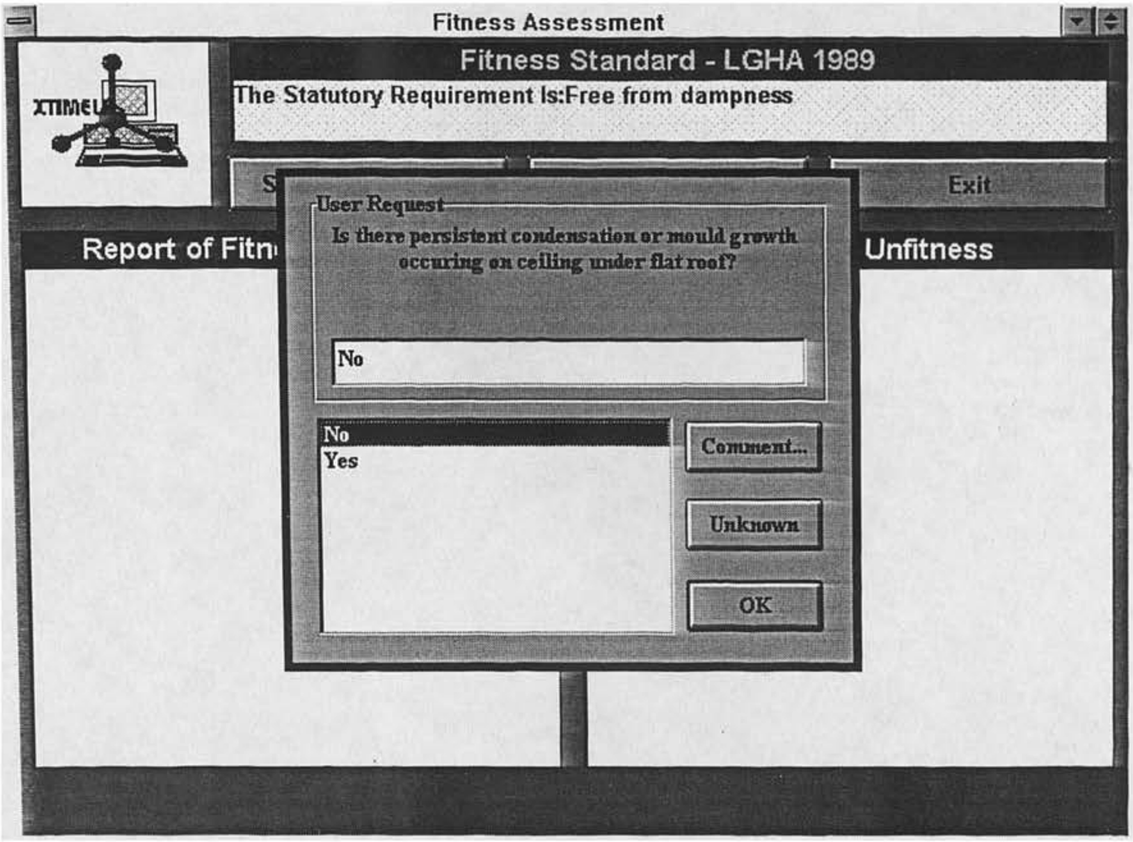
Screen 116: Input information about the condition of the building components



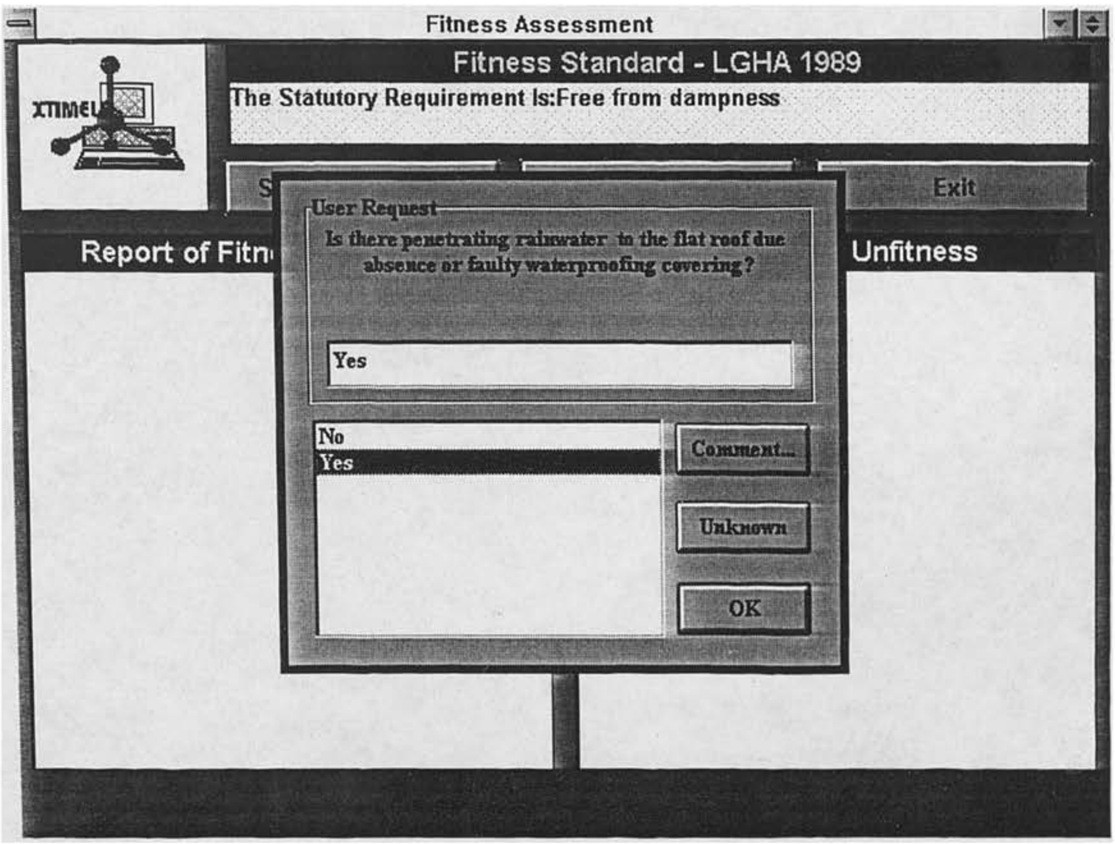
Screen 117: Input information about the condition of the building components



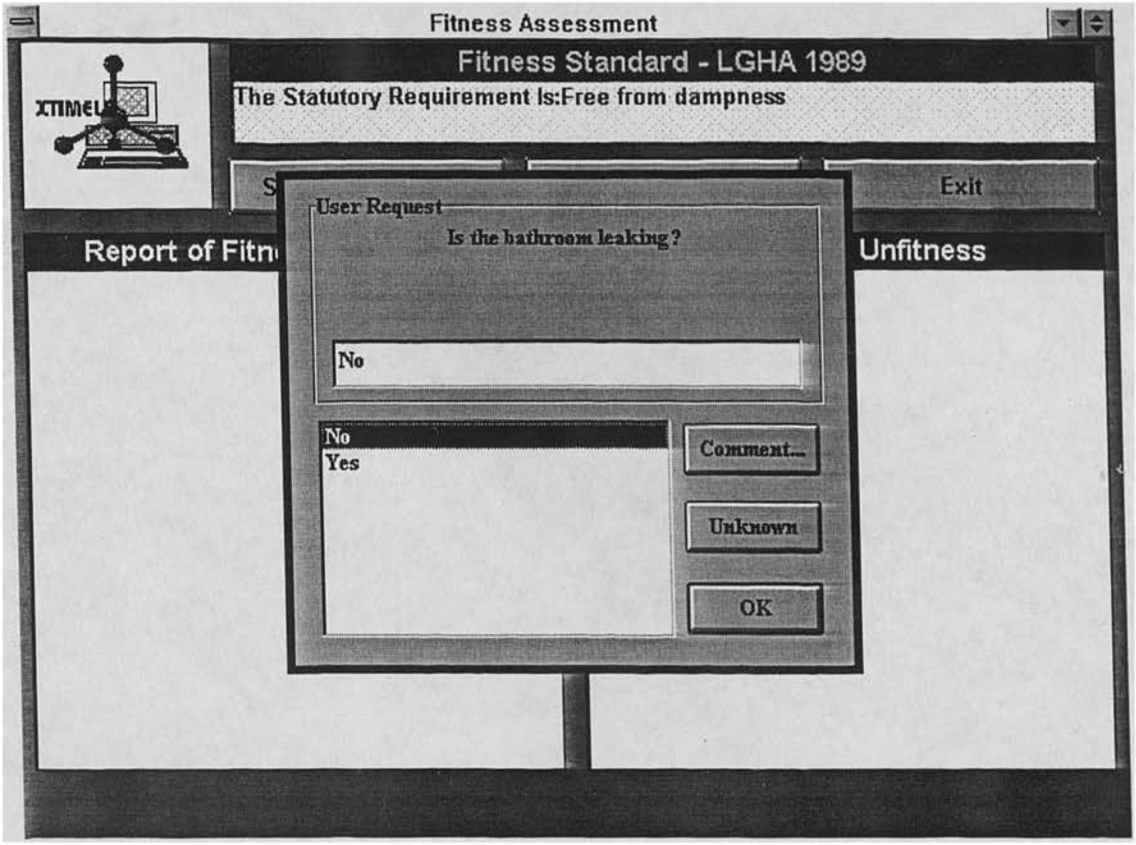
Screen 118: Input information about the condition of the building components



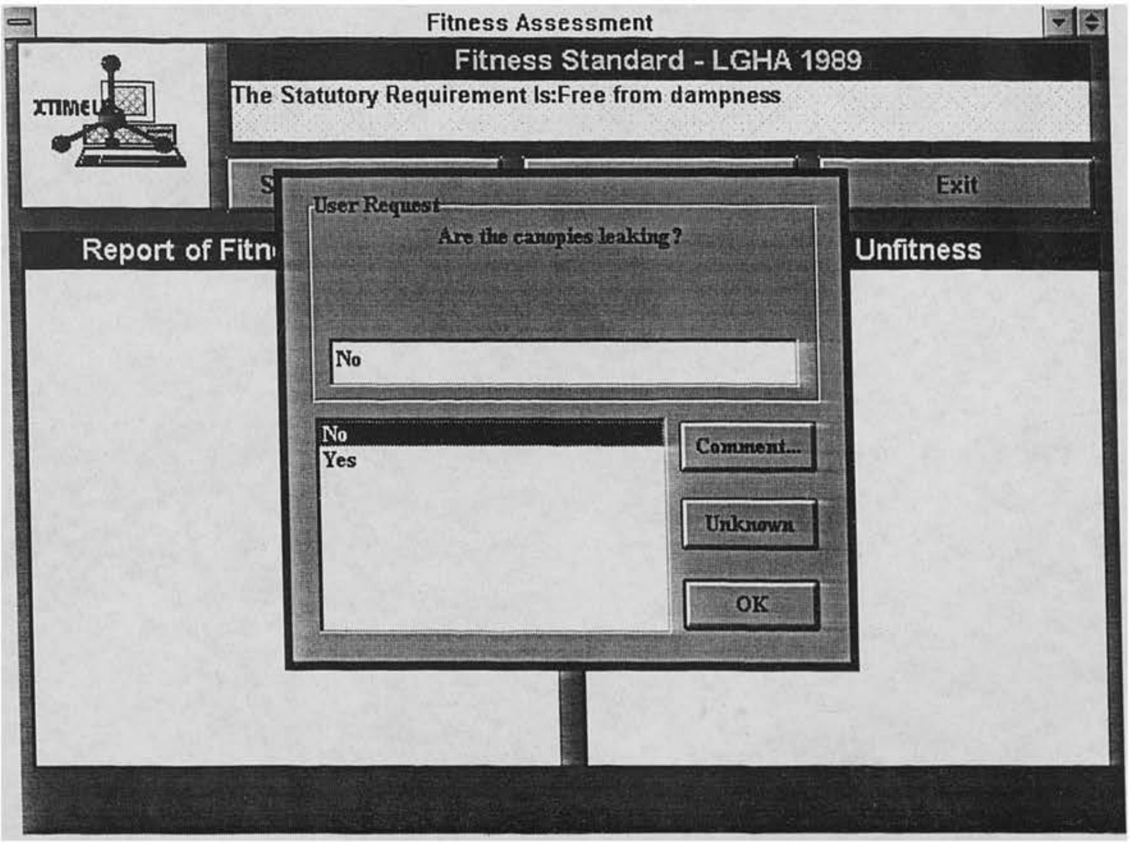
Screen 119: Input information about the condition of the building components



Screen 120: Input information about the condition of the building components

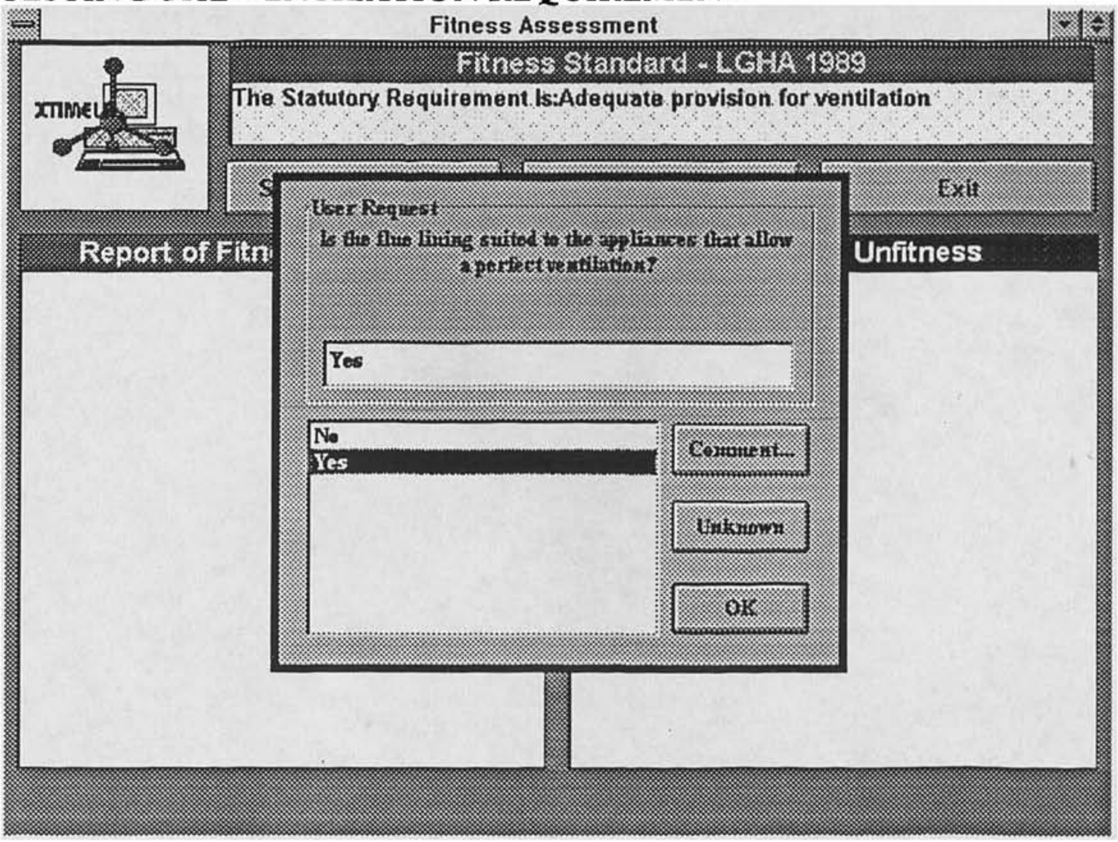


Screen 121: Input information about the condition of the building components

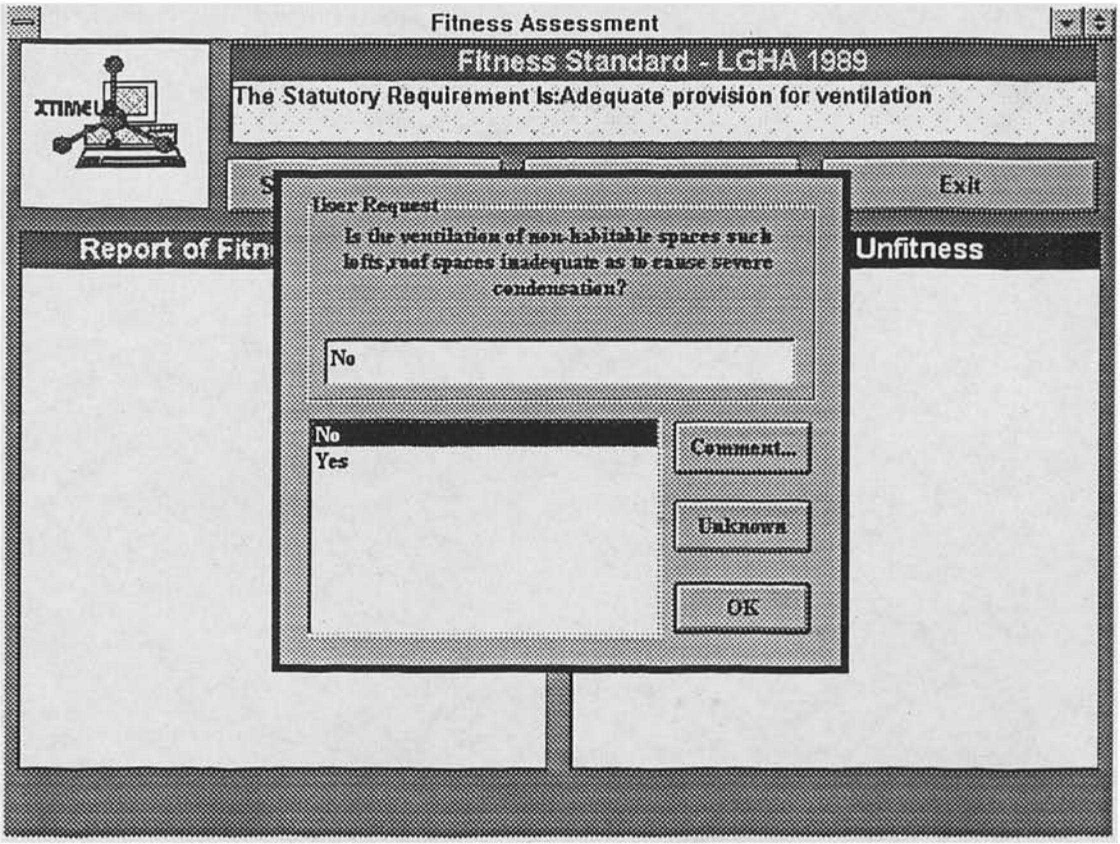


Screen 122: Input information about the condition of the building components

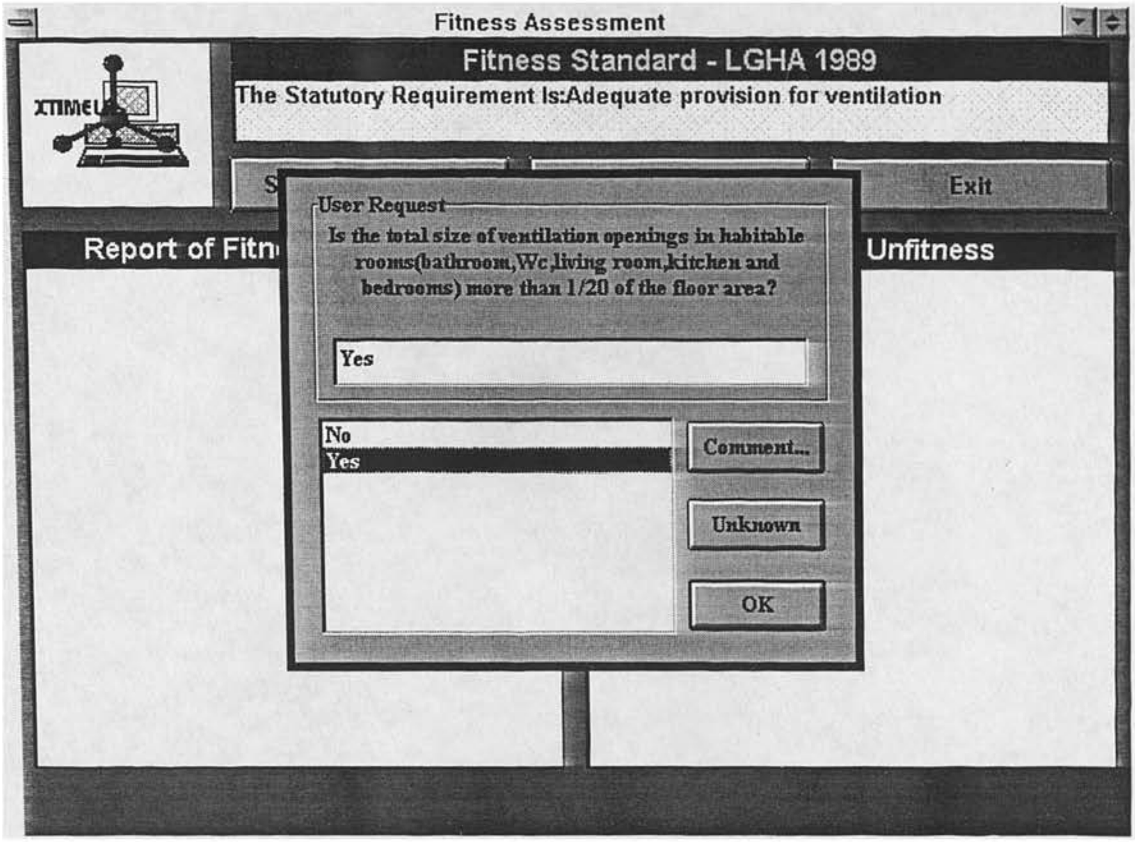
ASSESSING THE VENTILATION REQUIREMENT



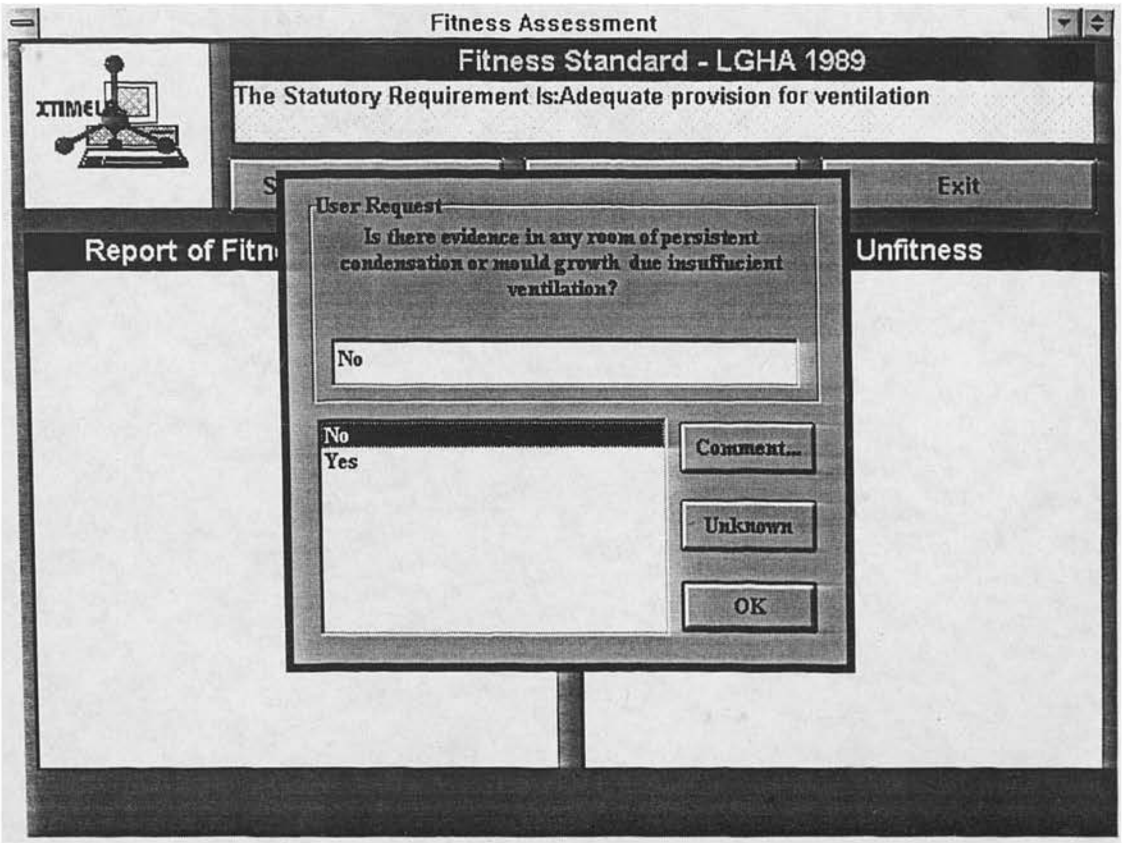
Screen 123: Input information about the condition of the building components



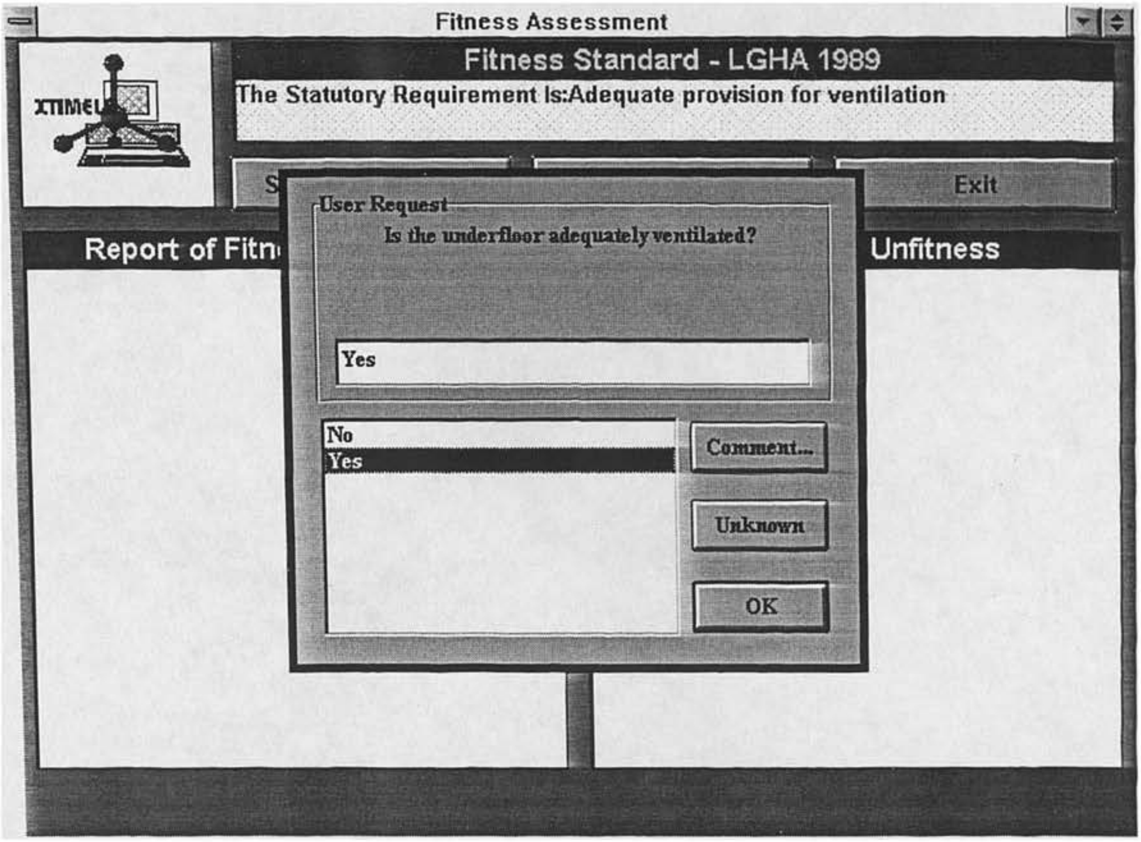
Screen 124: Input information about the condition of the building components



Screen 125: Input information about the condition of the building components

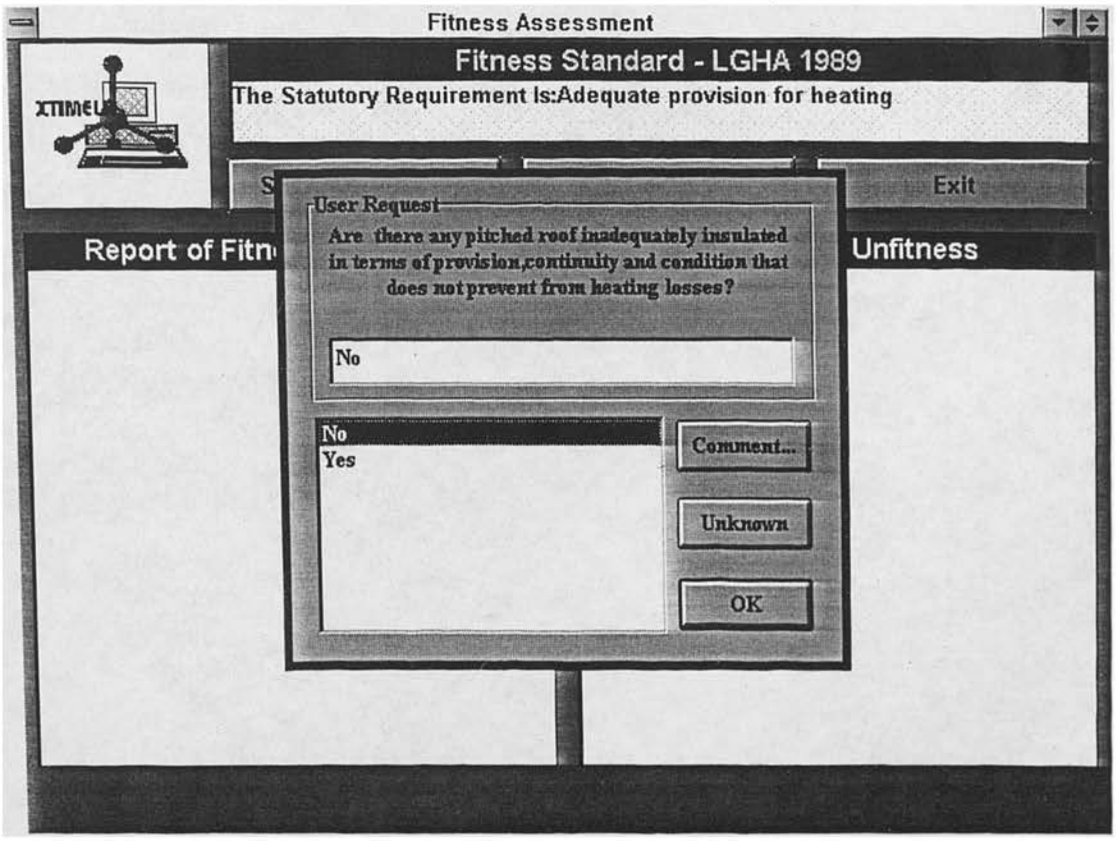


Screen 126: Input information about the condition of the building components

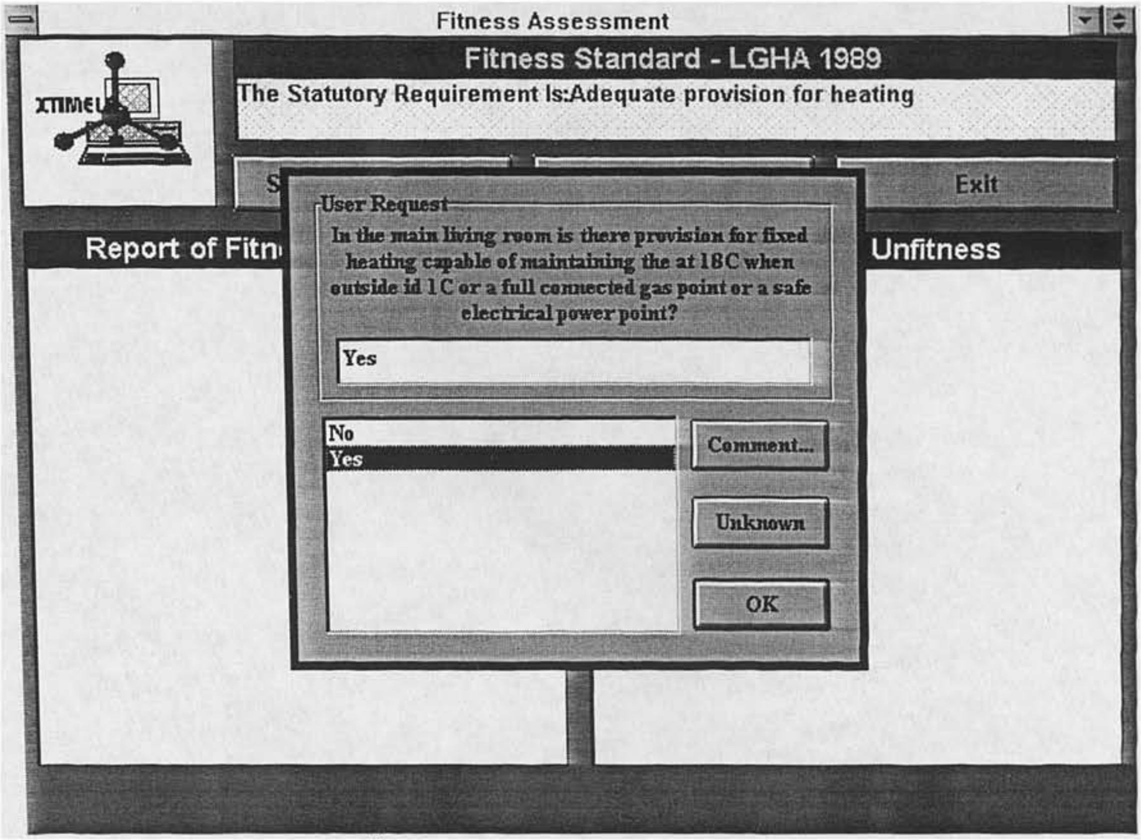


Screen 127: Input information about the condition of the building components

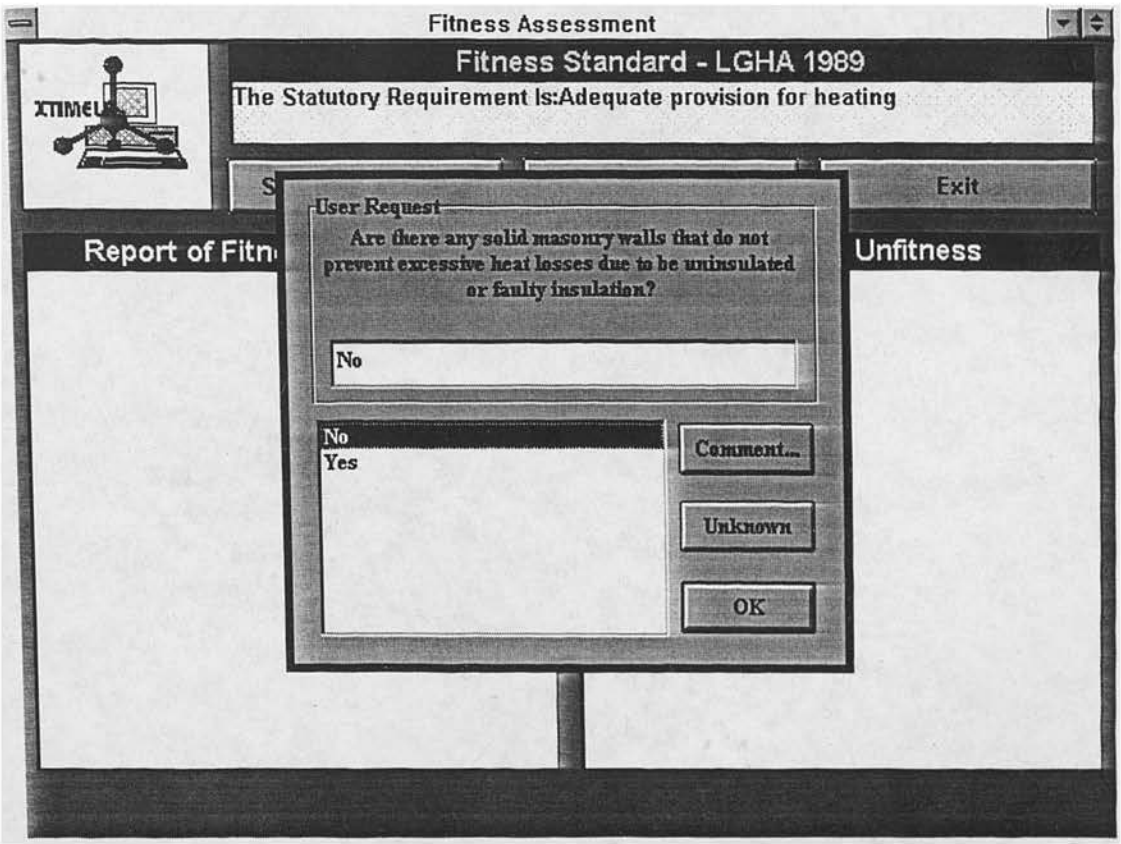
ASSESSING HEATING REQUIREMENT



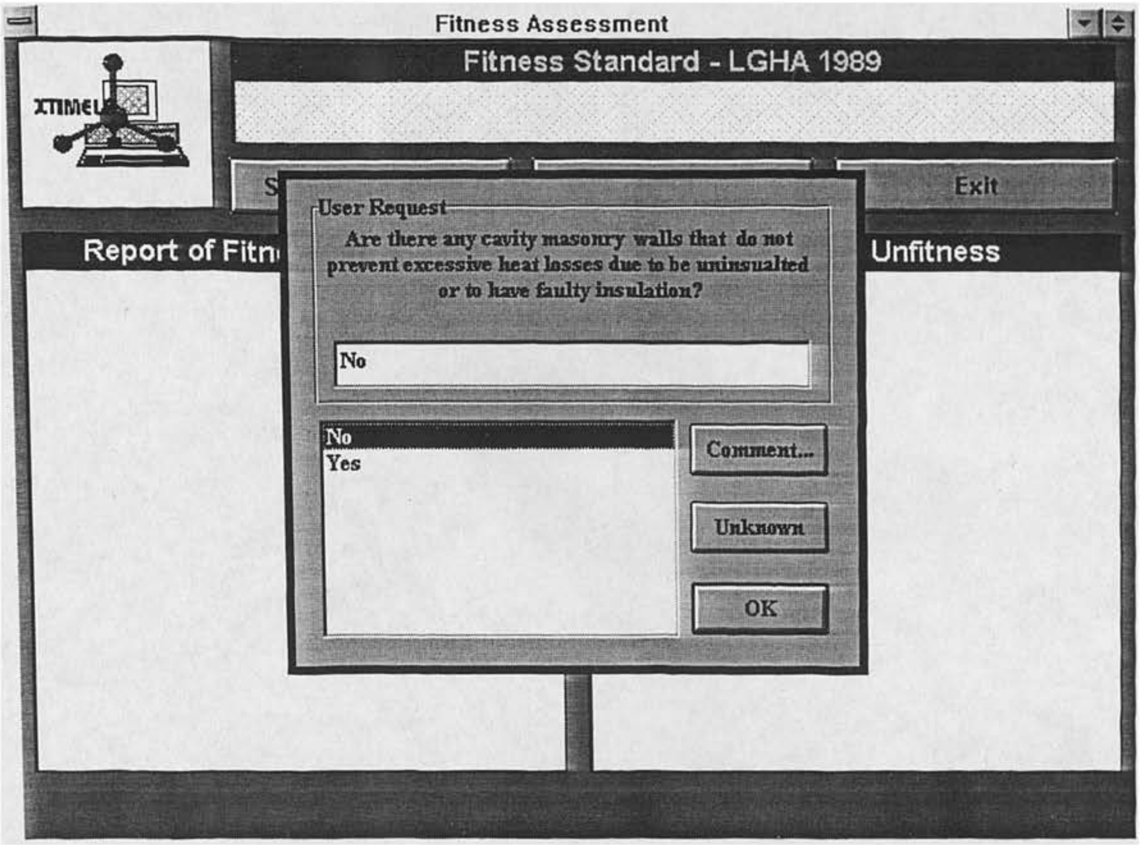
Screen 128: Input information about the condition of the building components



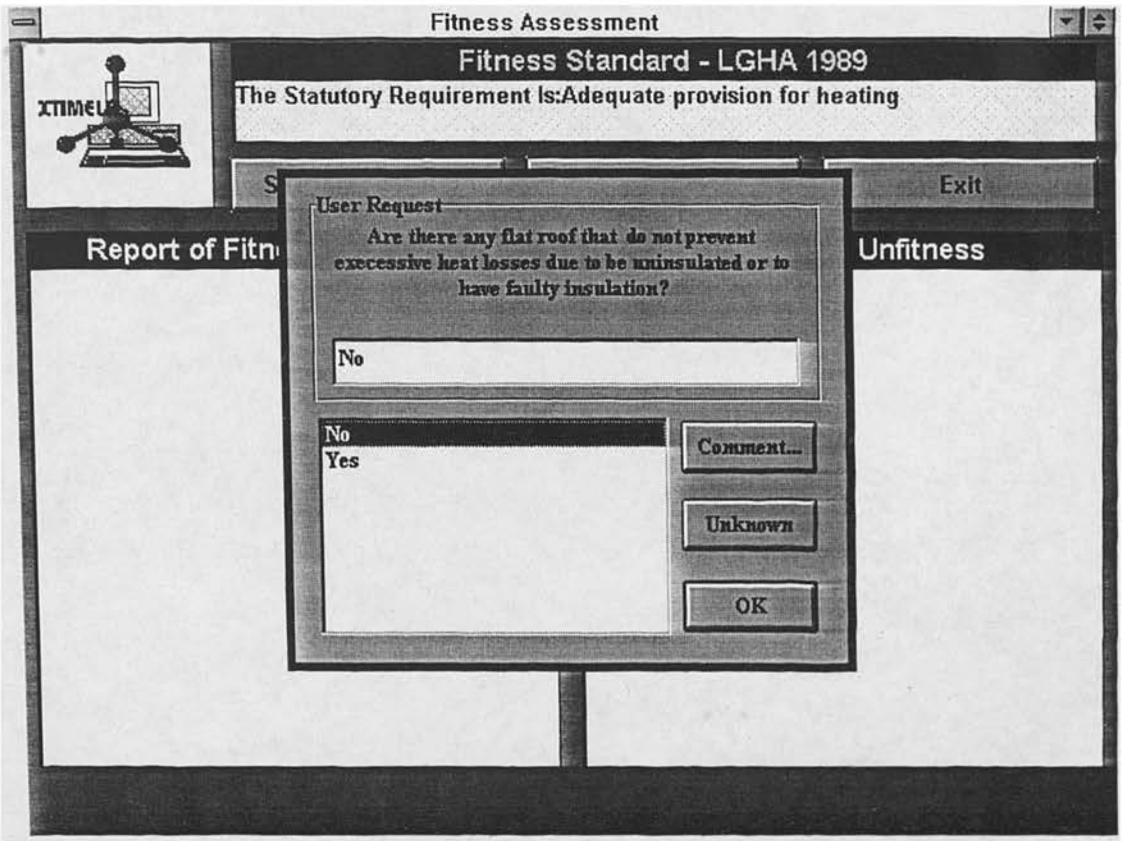
Screen 129: Input information about the condition of the building components



Screen 130: Input information about the condition of the building components

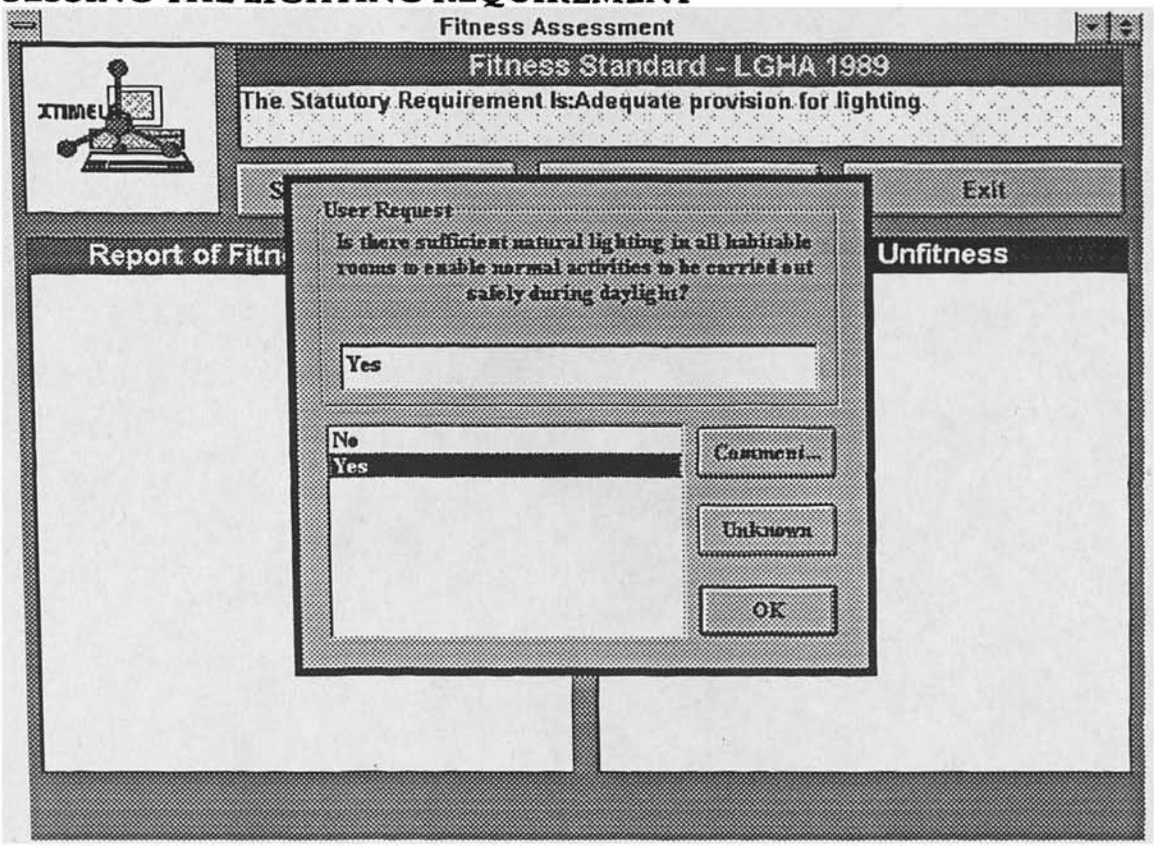


Screen 131: Input information about the condition of the building components

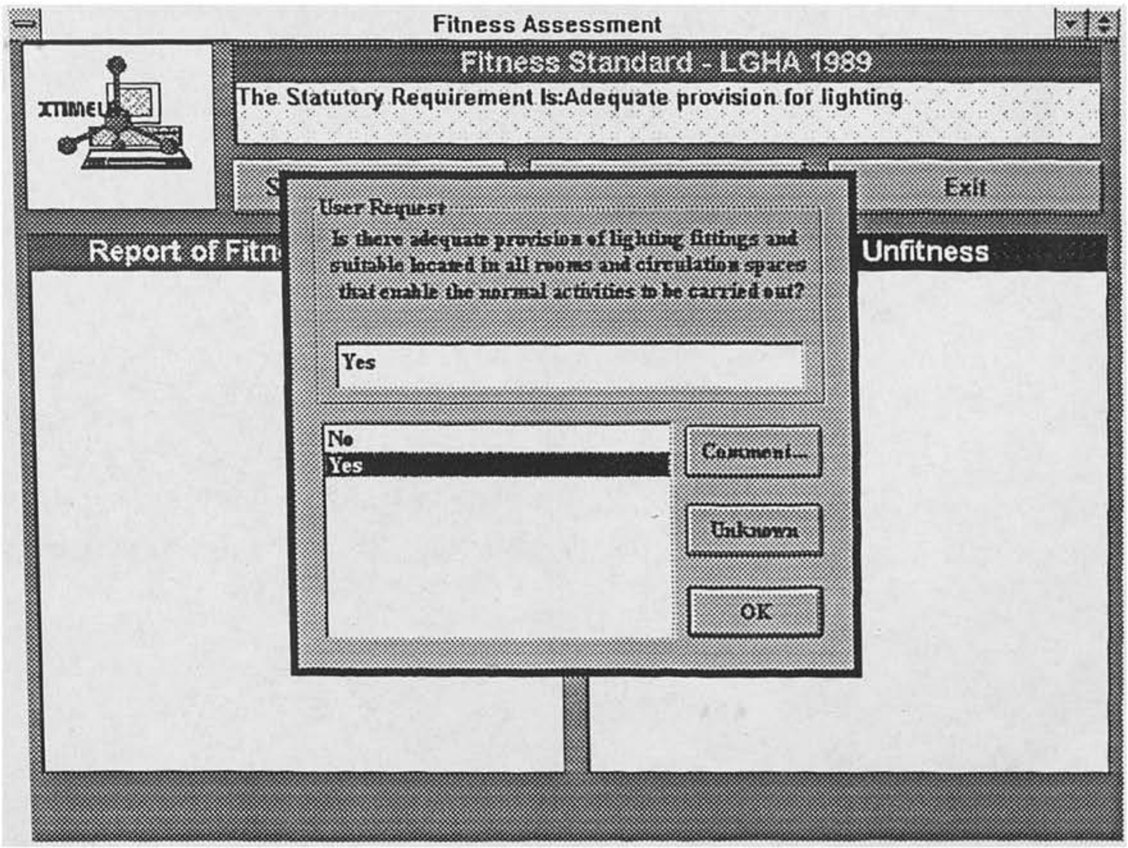


Screen 132: Input information about the condition of the building components

ASSESSING THE LIGHTING REQUIREMENT

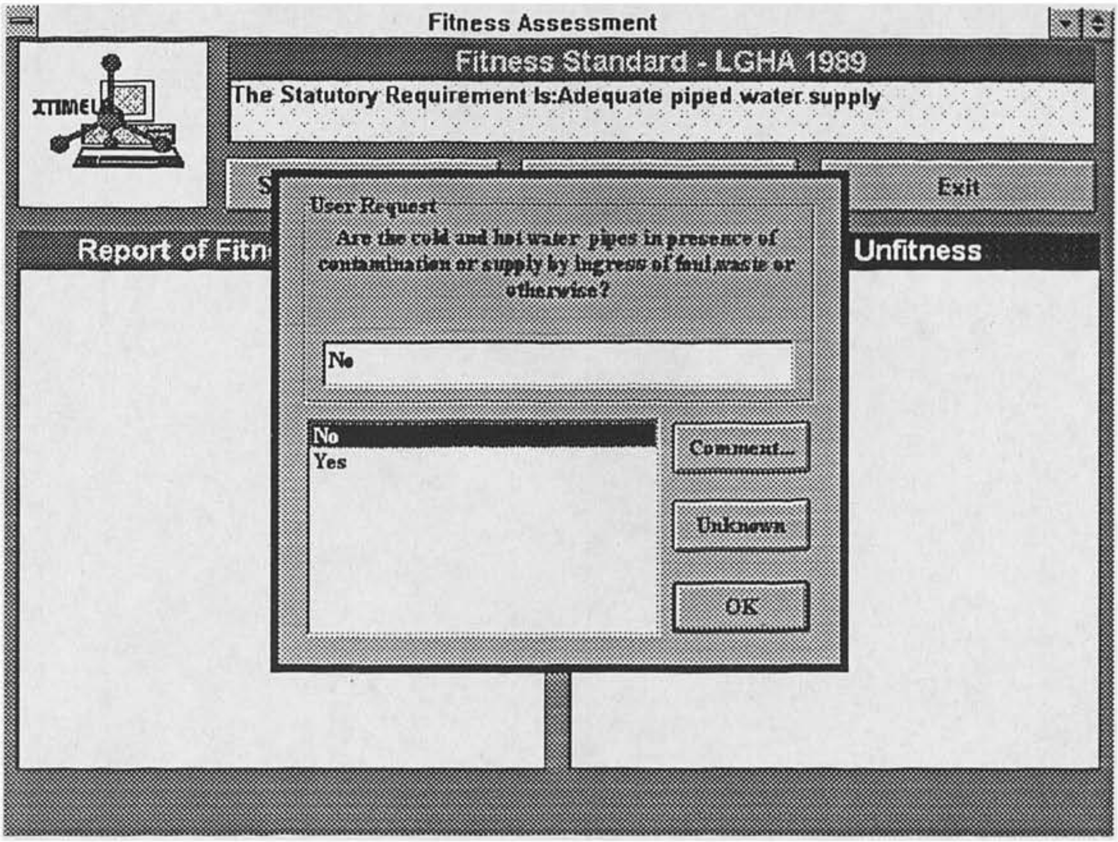


Screen 133: Input information about the condition of the building components

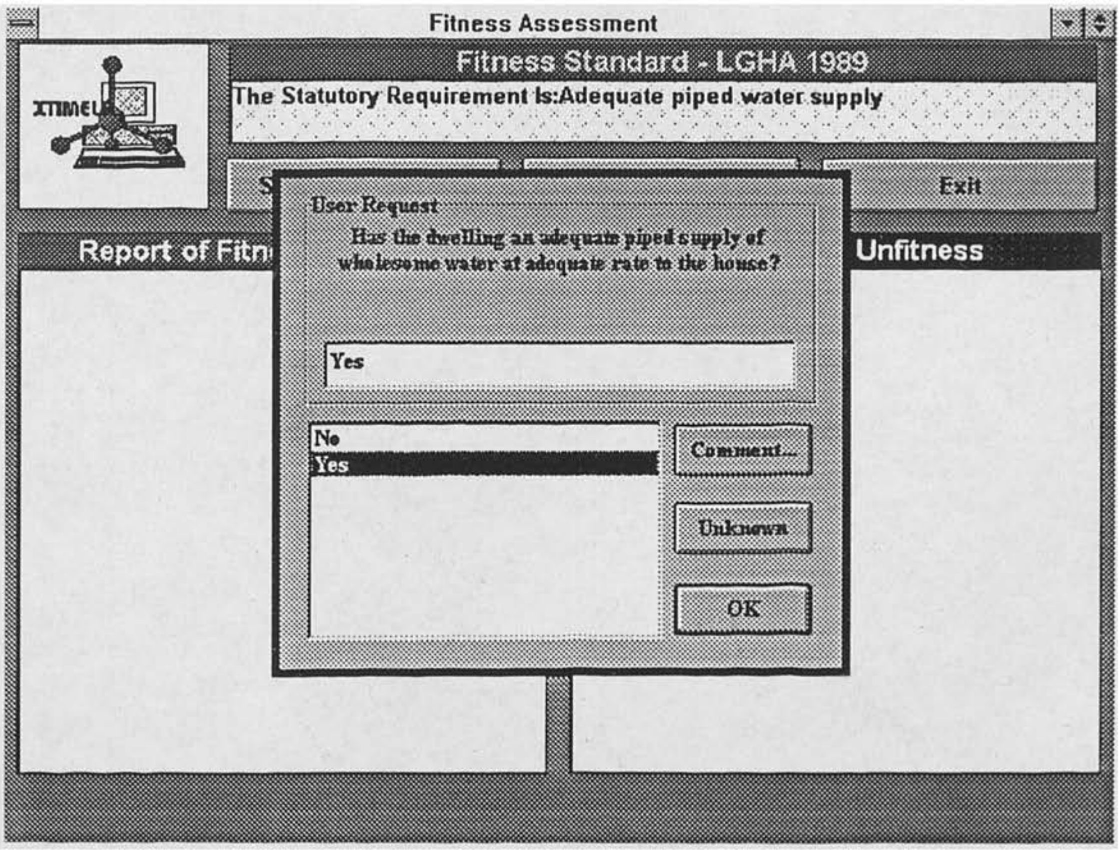


Screen 134: Input information about the condition of the building components

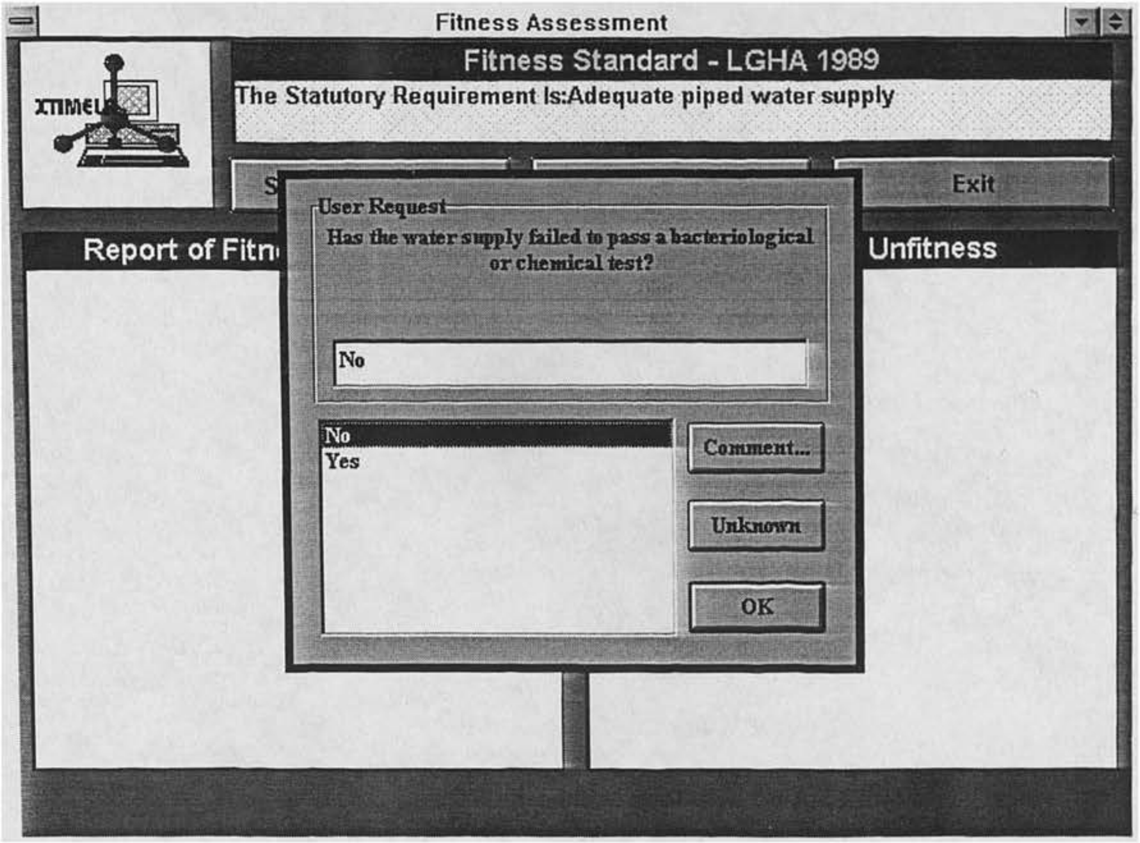
ASSESSING THE WATER SUPPLY REQUIREMENT



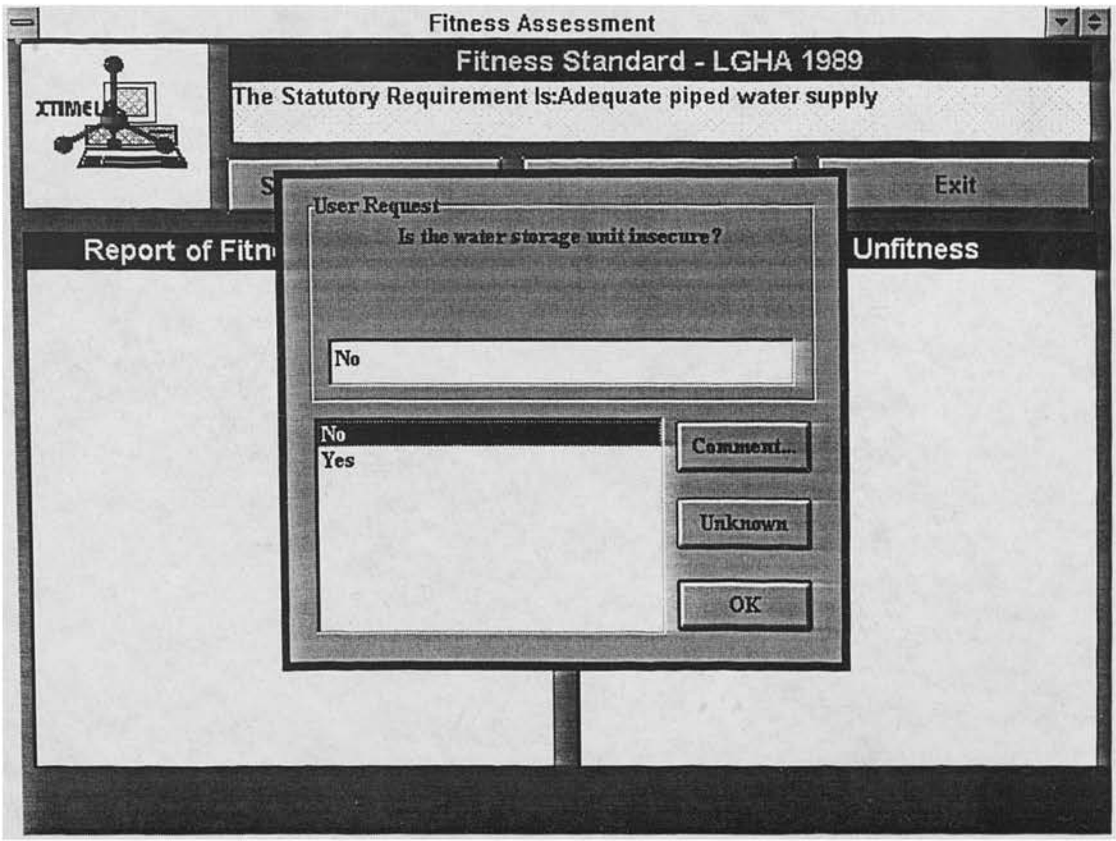
Screen 135: Input information about the condition of the building components



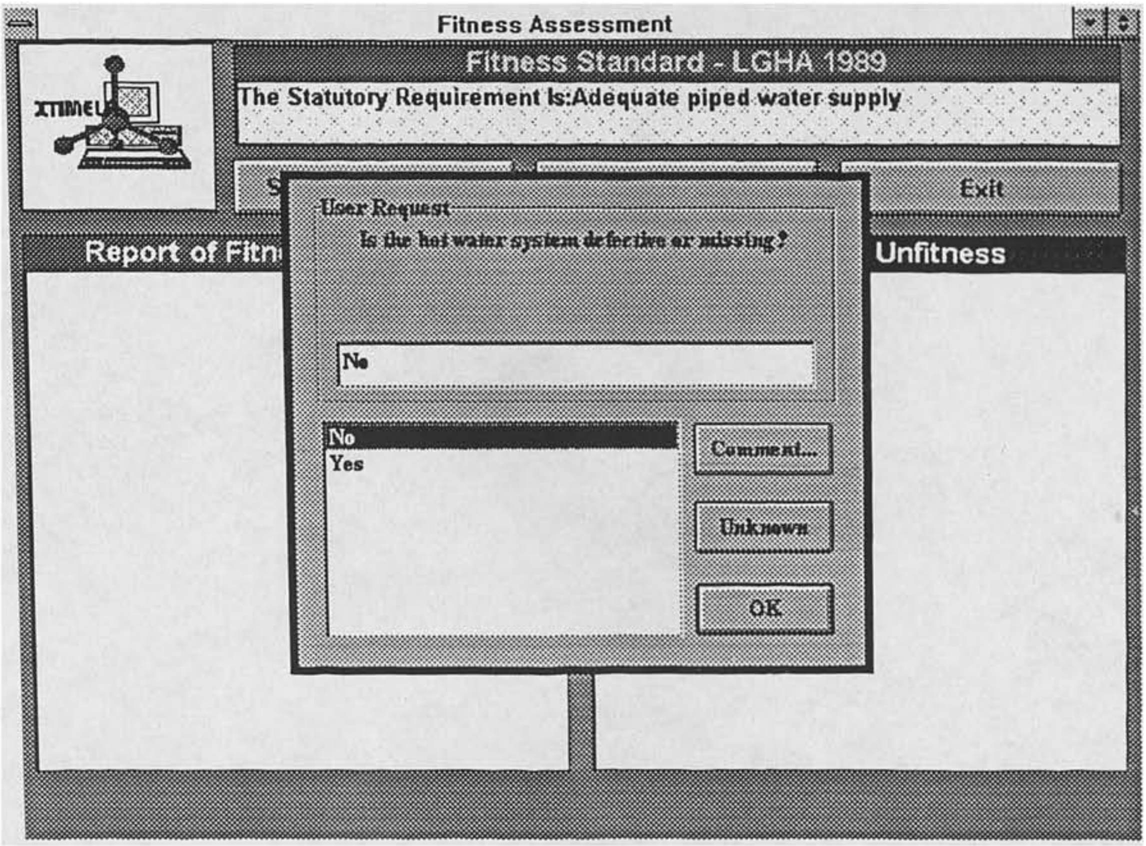
Screen 136: Input information about the condition of the building components



Screen 137: Input information about the condition of the building components

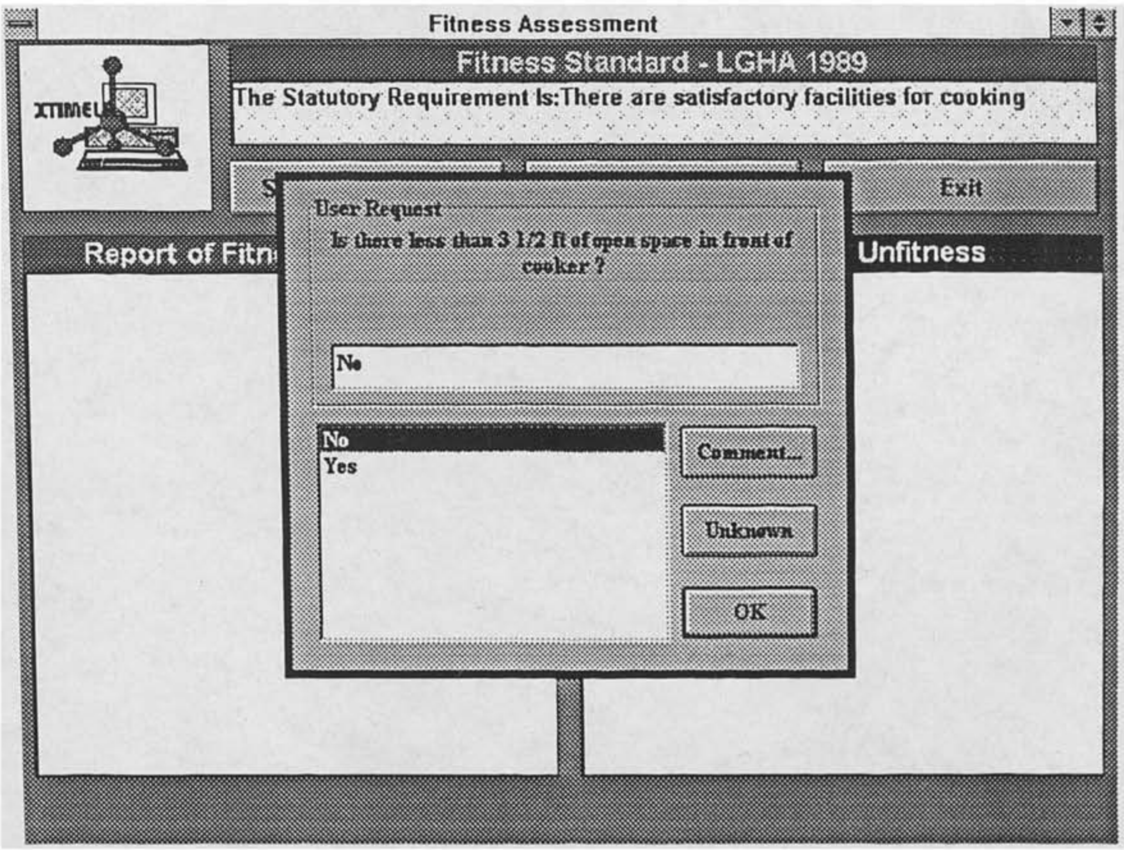


Screen 138: Input information about the condition of the building components

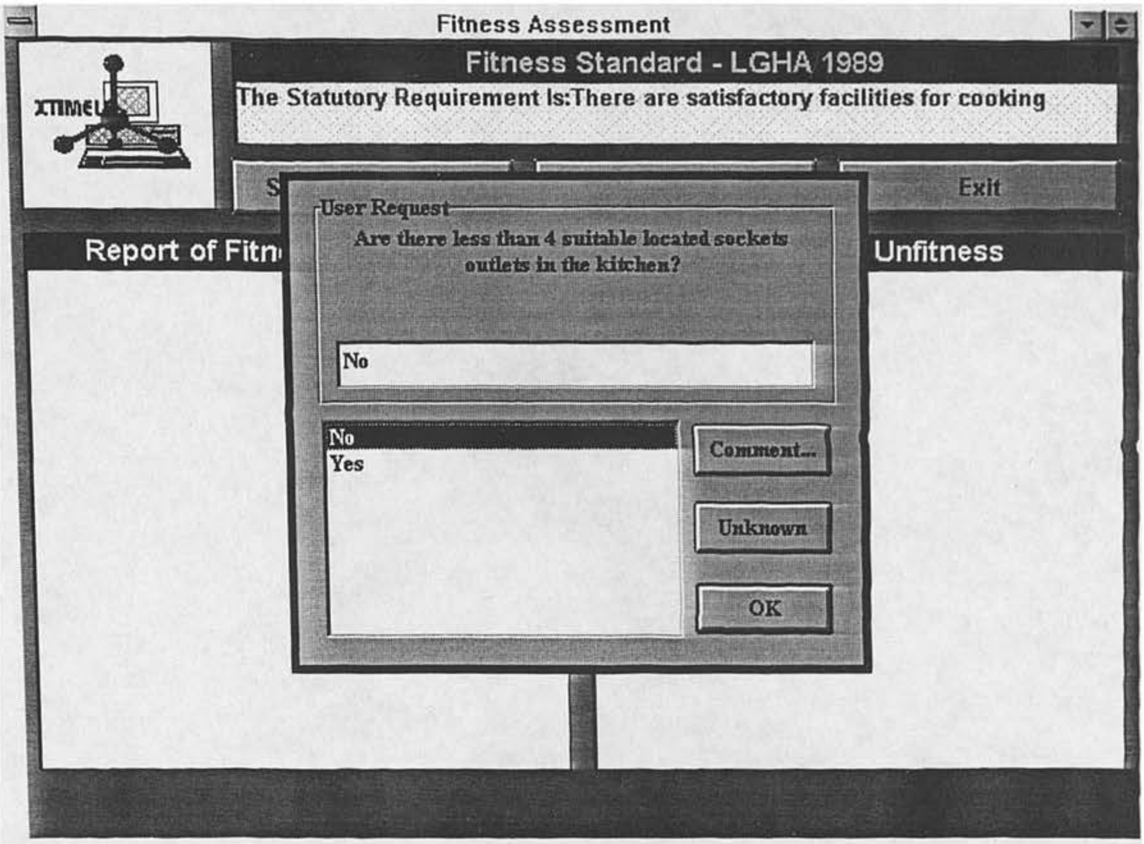


Screen 139: Input information about the condition of the building components

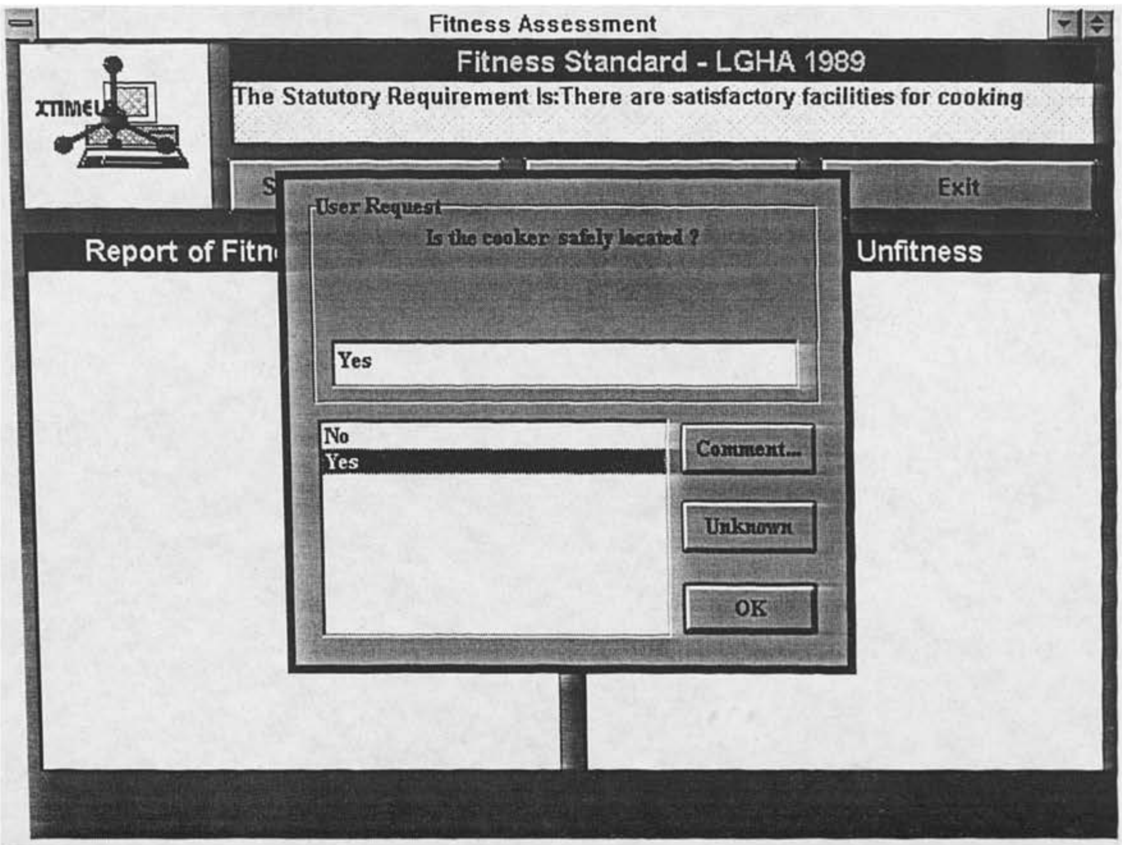
ASSESSING THE PREPARATION OF FOOD REQUIREMENT



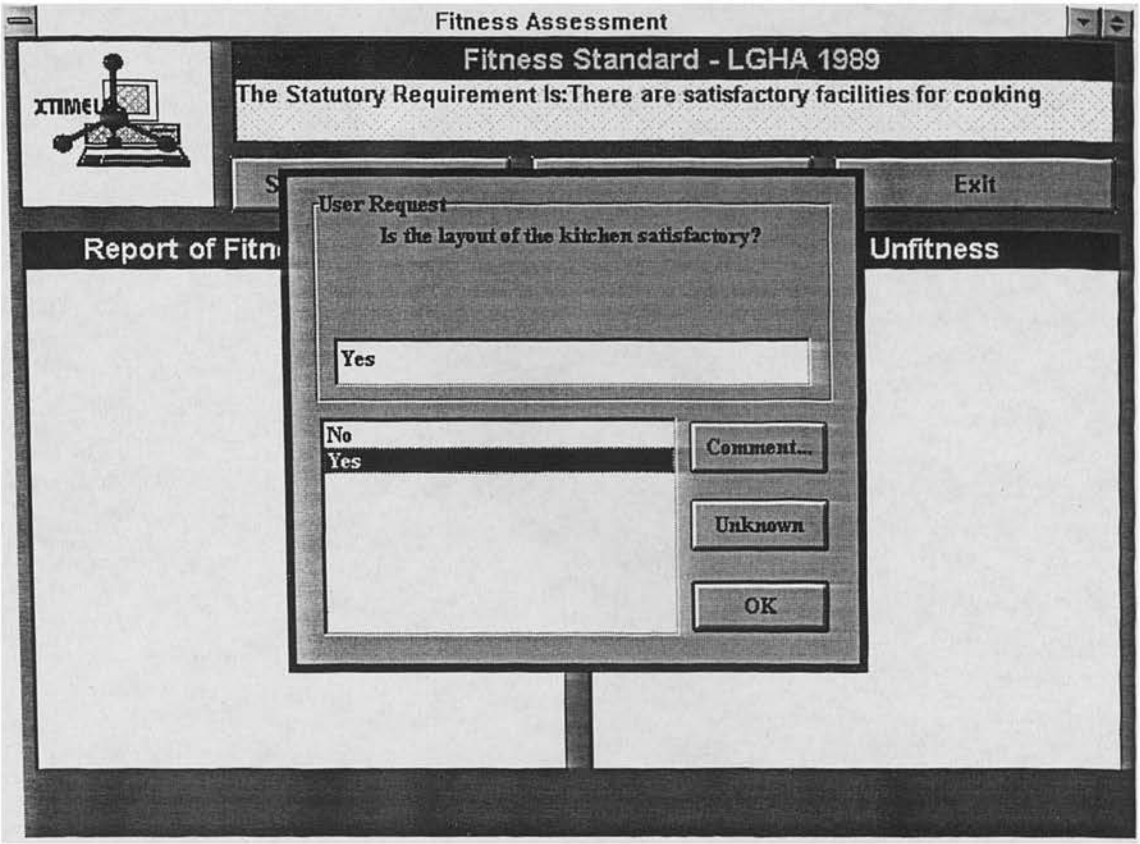
Screen 140: Input information about the condition of the building components



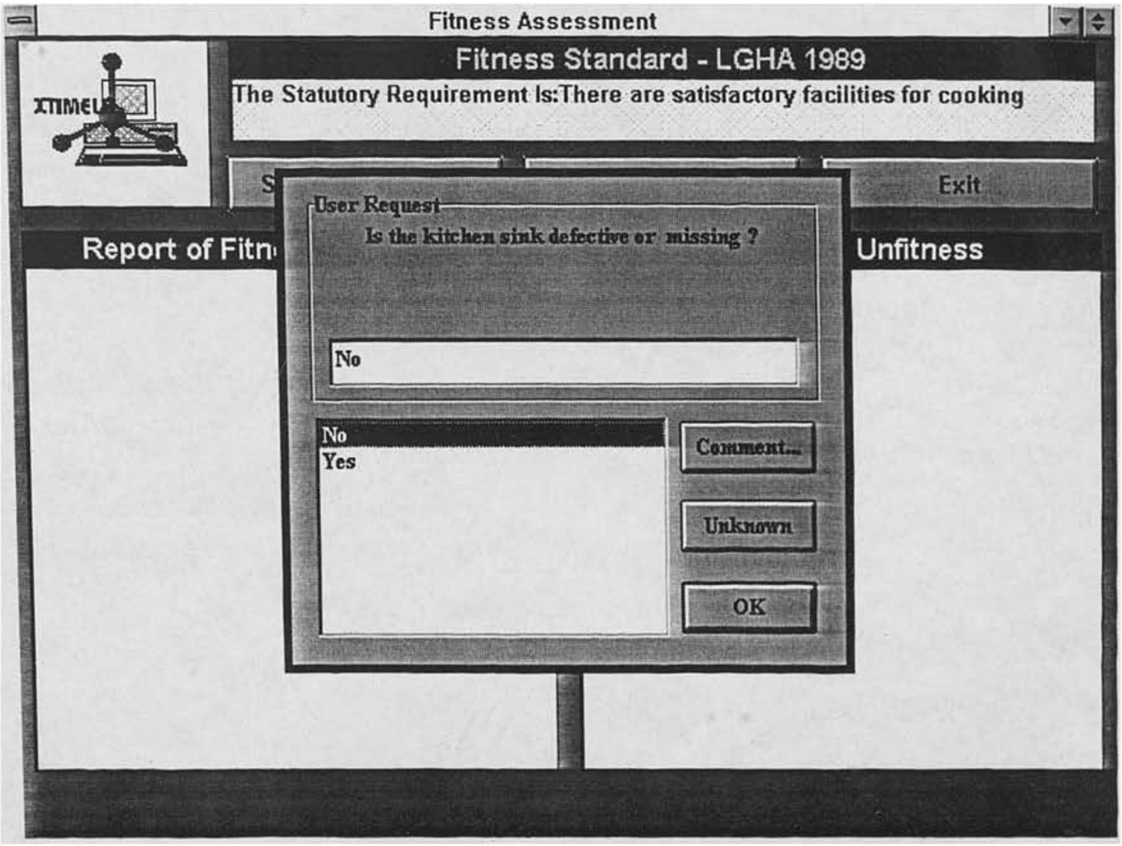
Screen 141: Input information about the condition of the building components



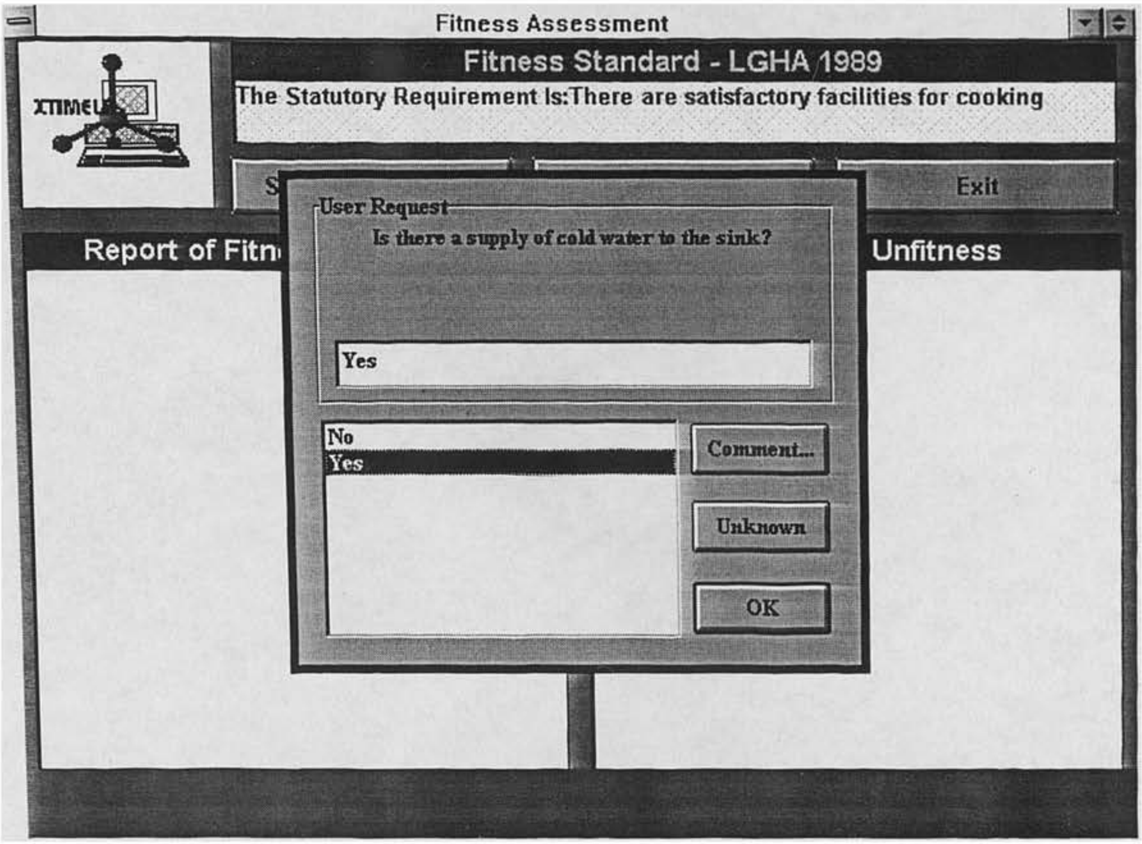
Screen 142: Input information about the condition of the building components



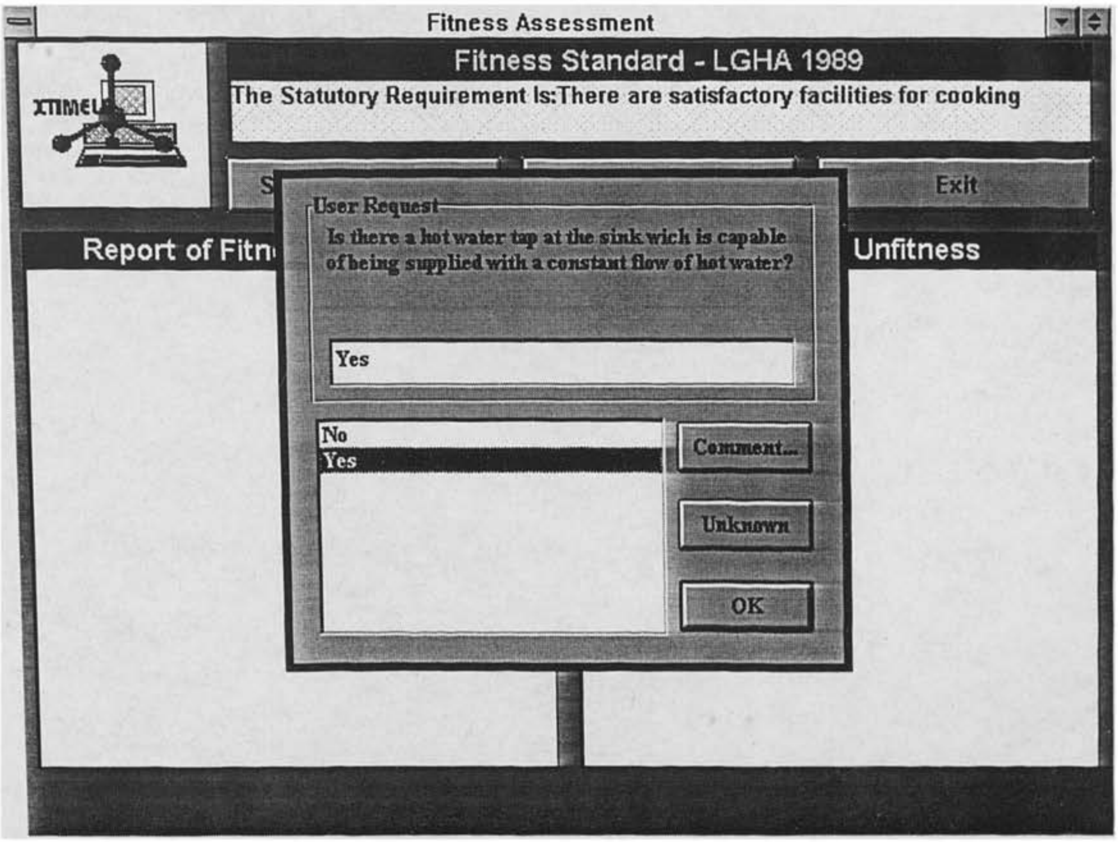
Screen 143: Input information about the condition of the building components



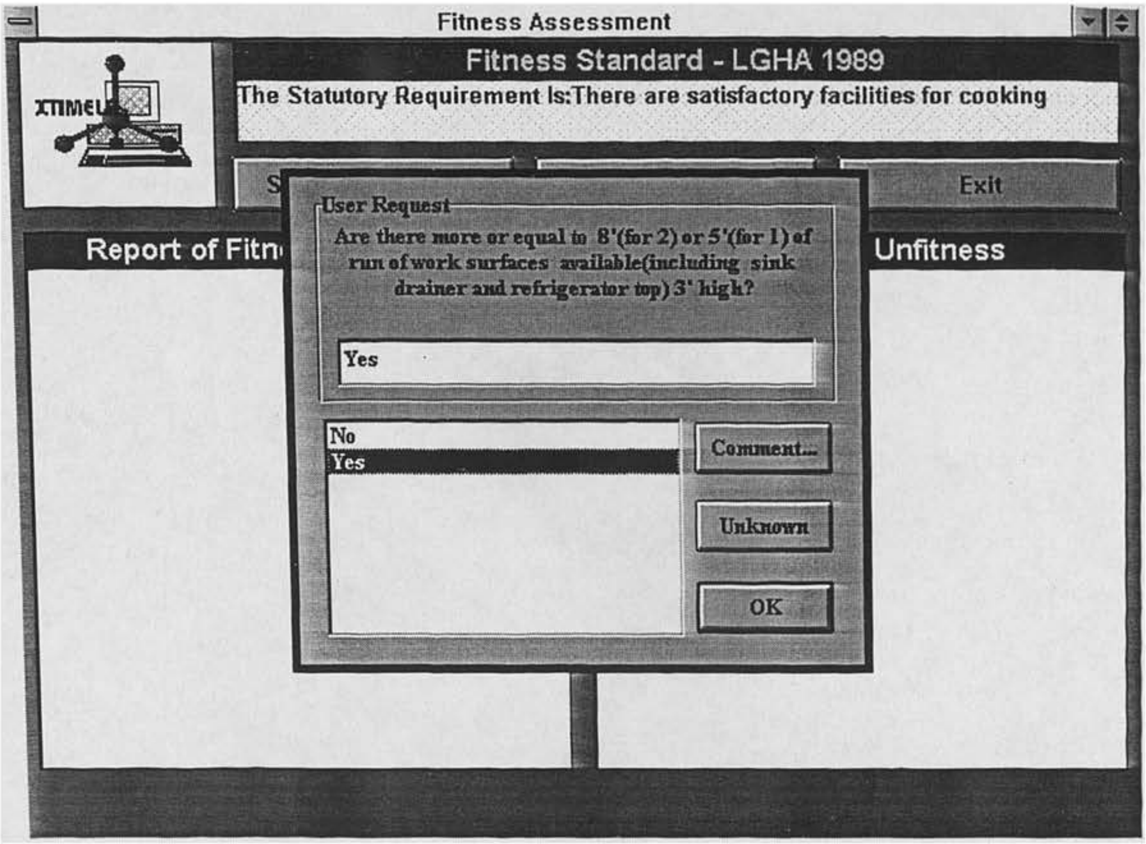
Screen 144: Input information about the condition of the building components



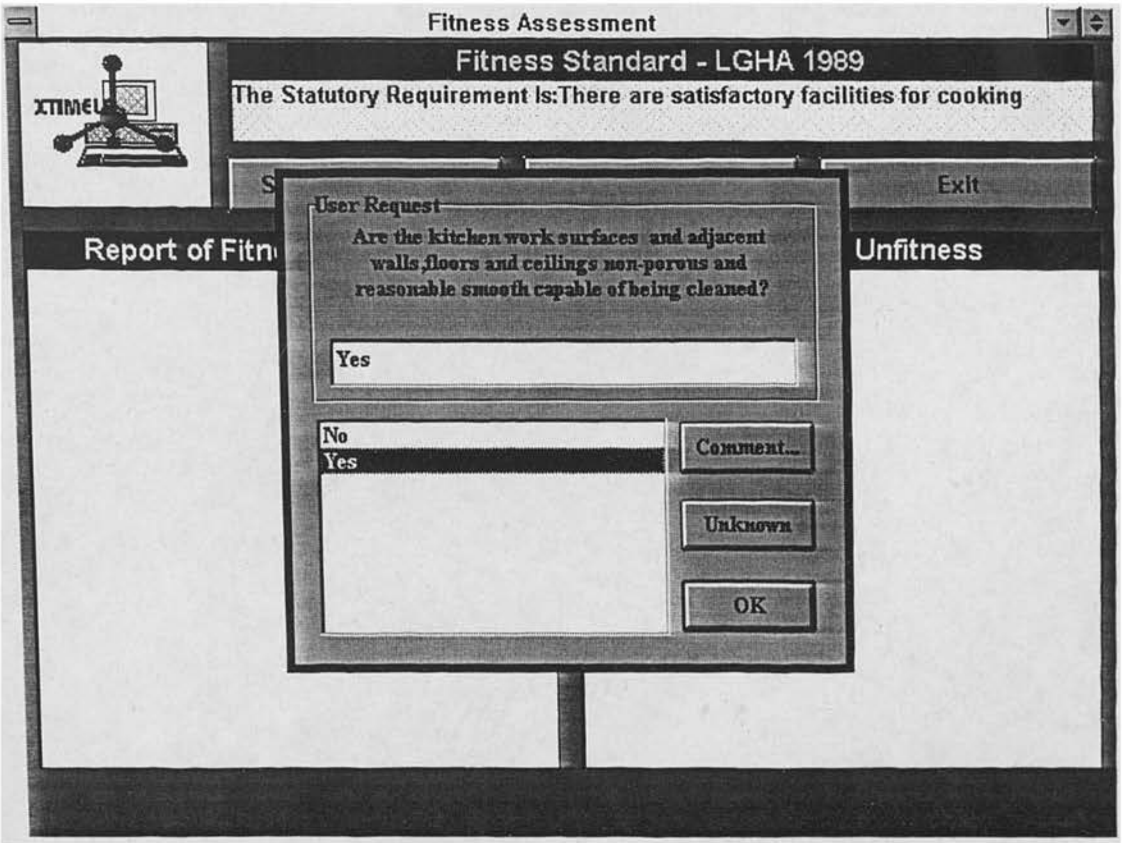
Screen 145: Input information about the condition of the building components



Screen 146: Input information about the condition of the building components

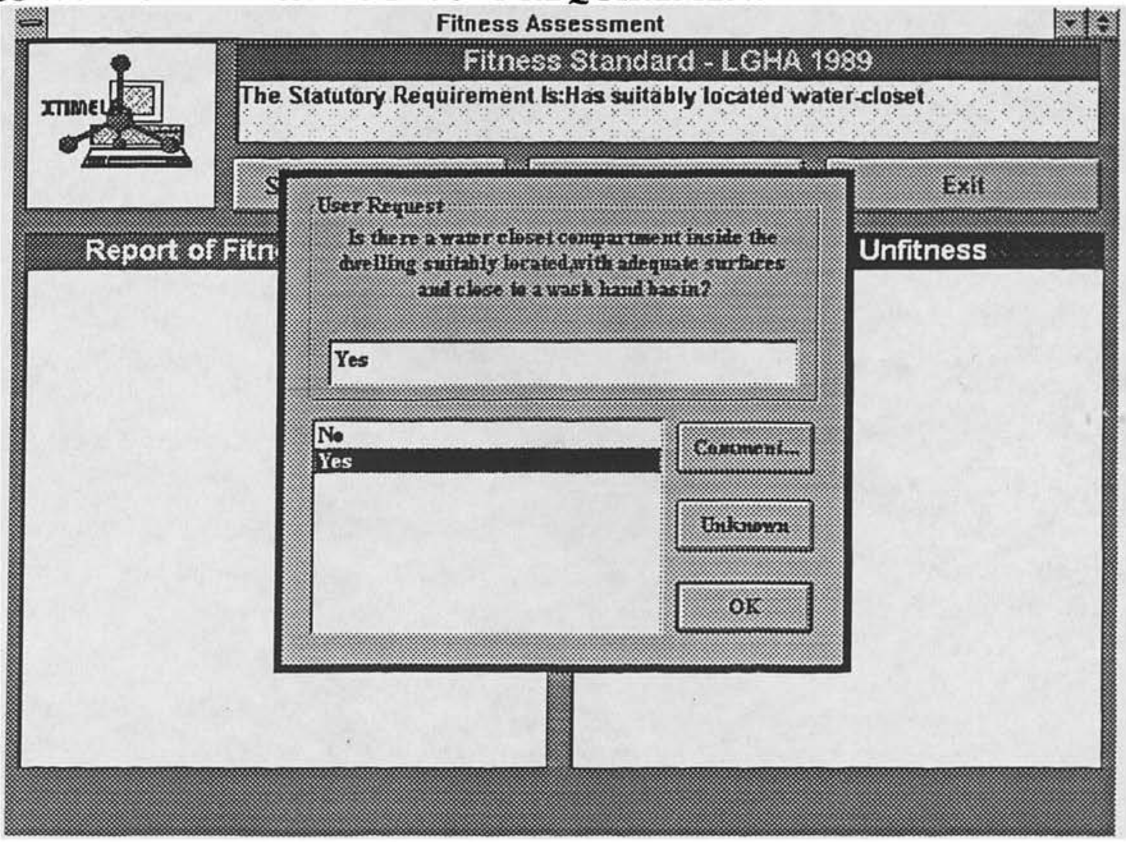


Screen 147: Input information about the condition of the building components



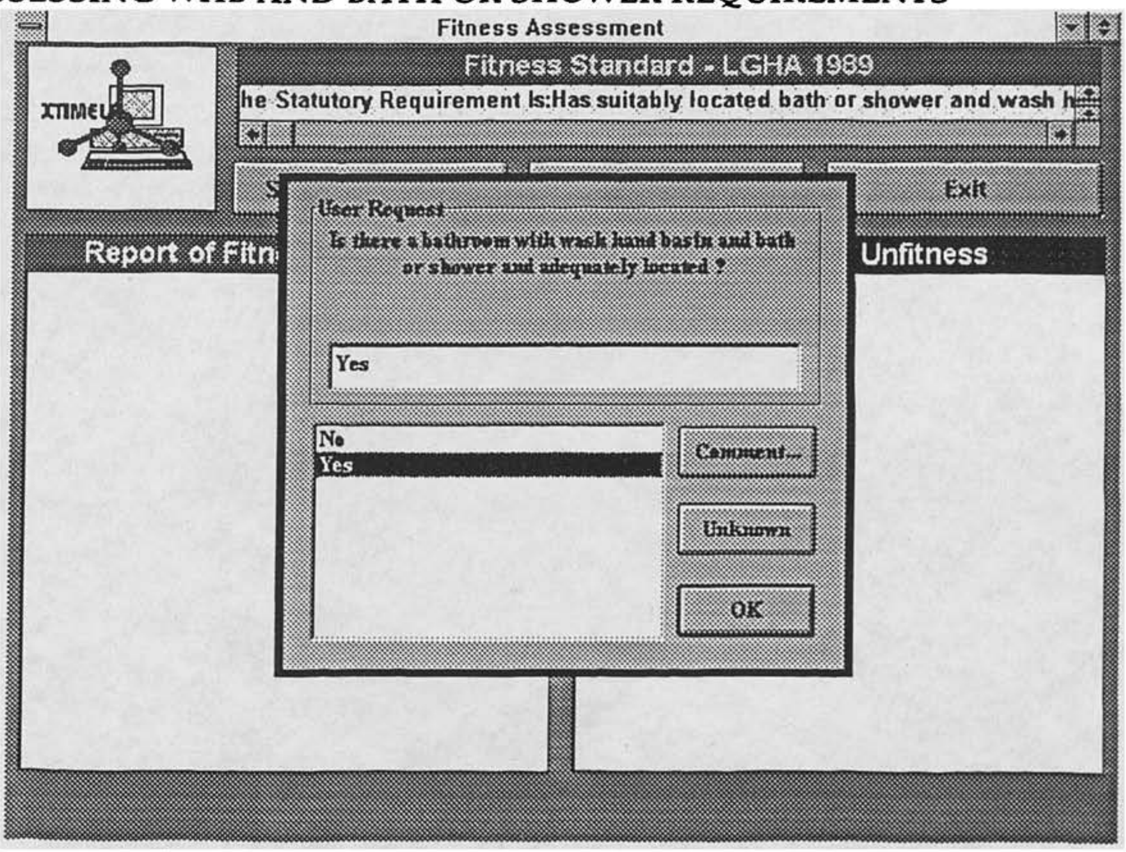
Screen 148: Input information about the condition of the building components

ASSESSING THE WATER CLOSET REQUIREMENT

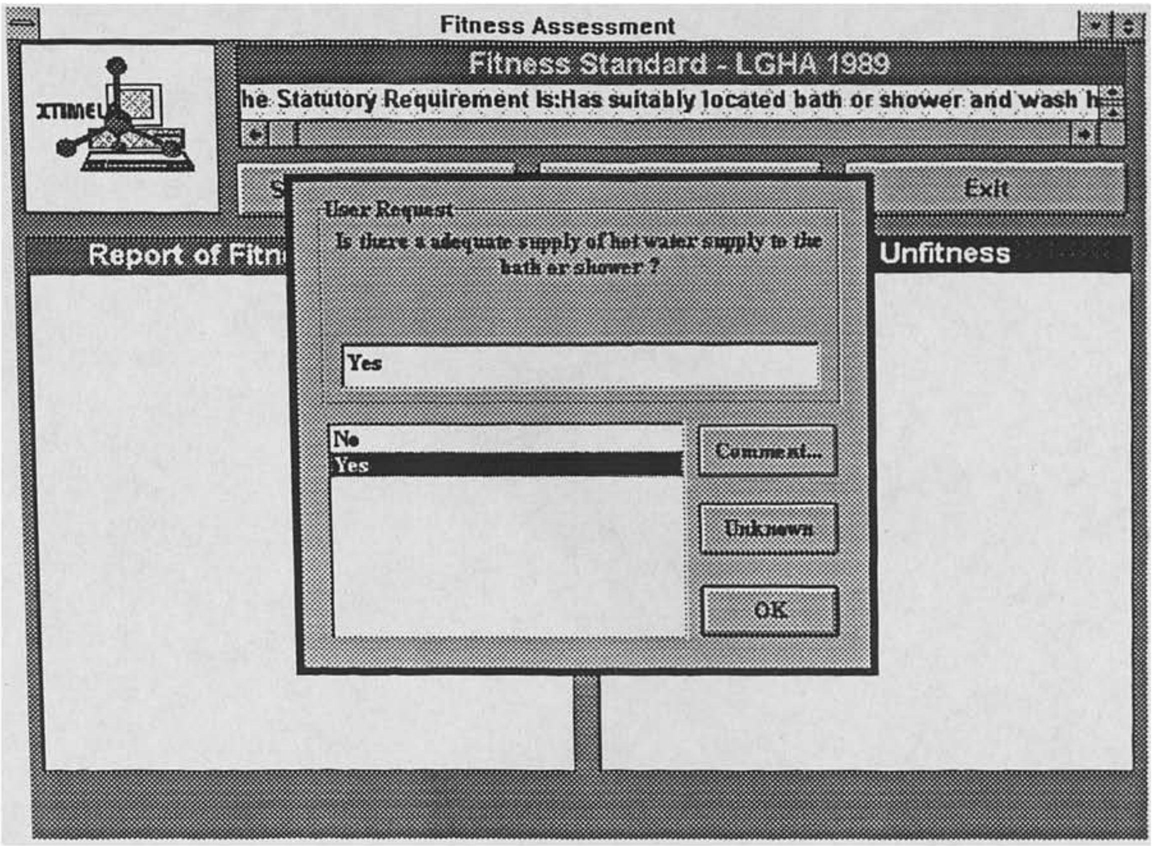


Screen 149: Input information about the condition of the building components

ASSESSING WHB AND BATH OR SHOWER REQUIREMENTS

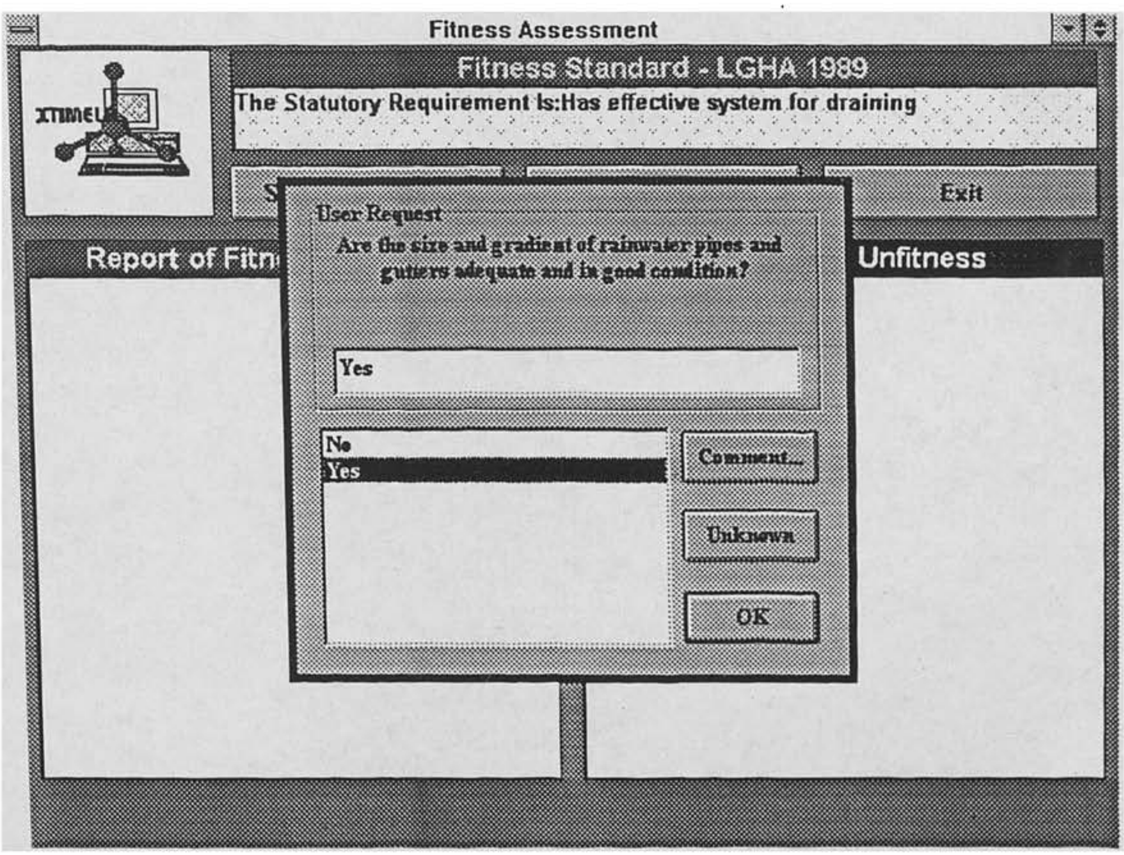


Screen 150: Input information about the condition of the building components

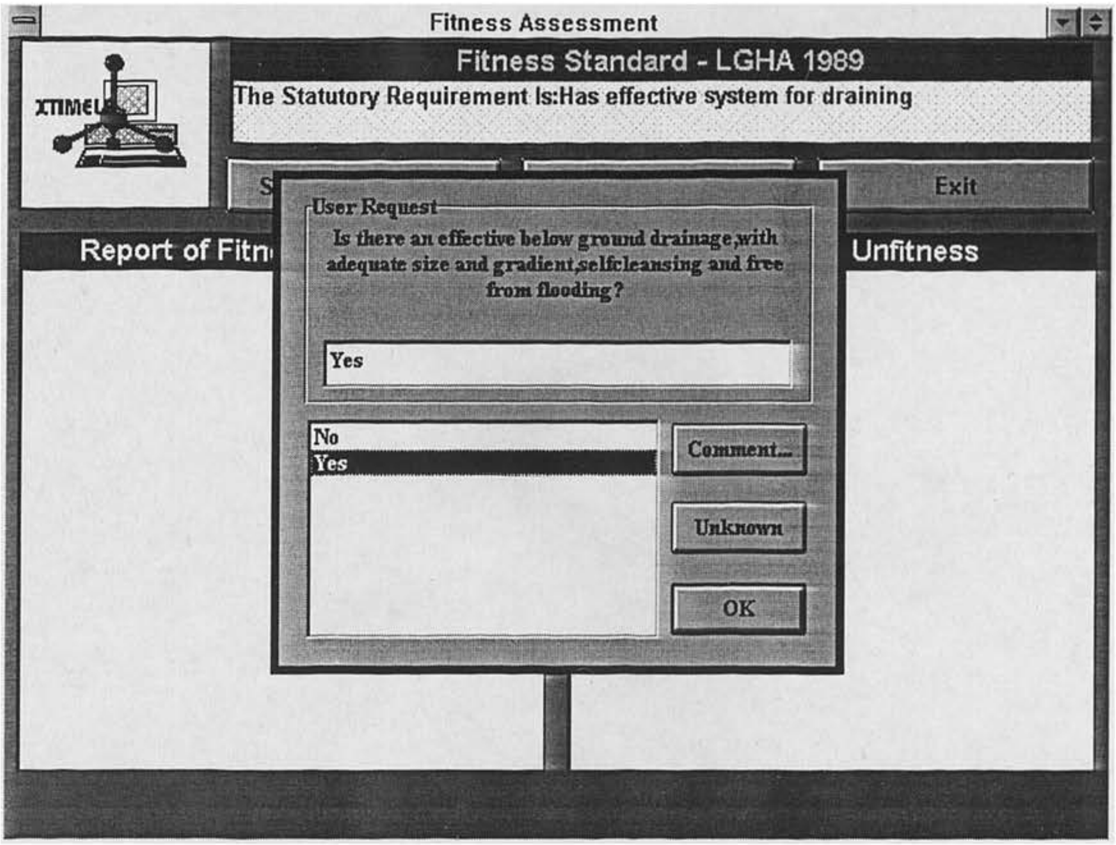


Screen 151: Input information about the condition of the building components

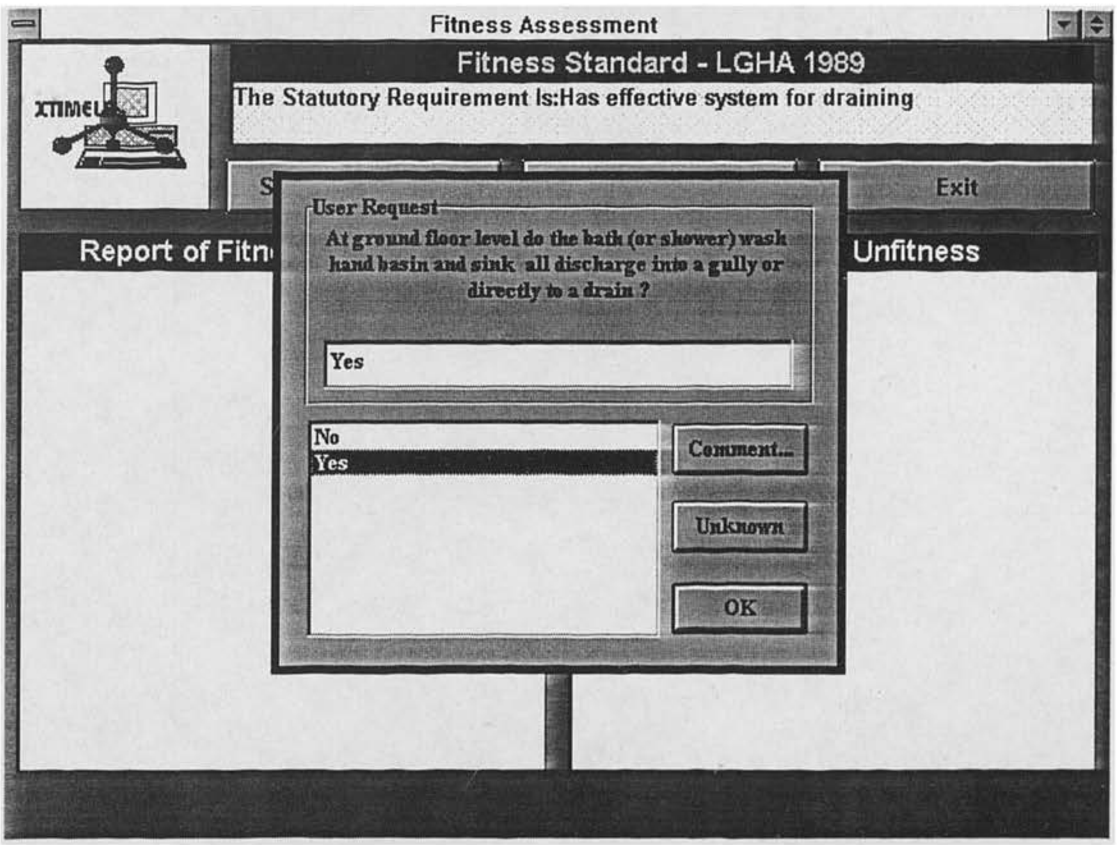
ASSESSING THE DRAINAGE REQUIREMENT



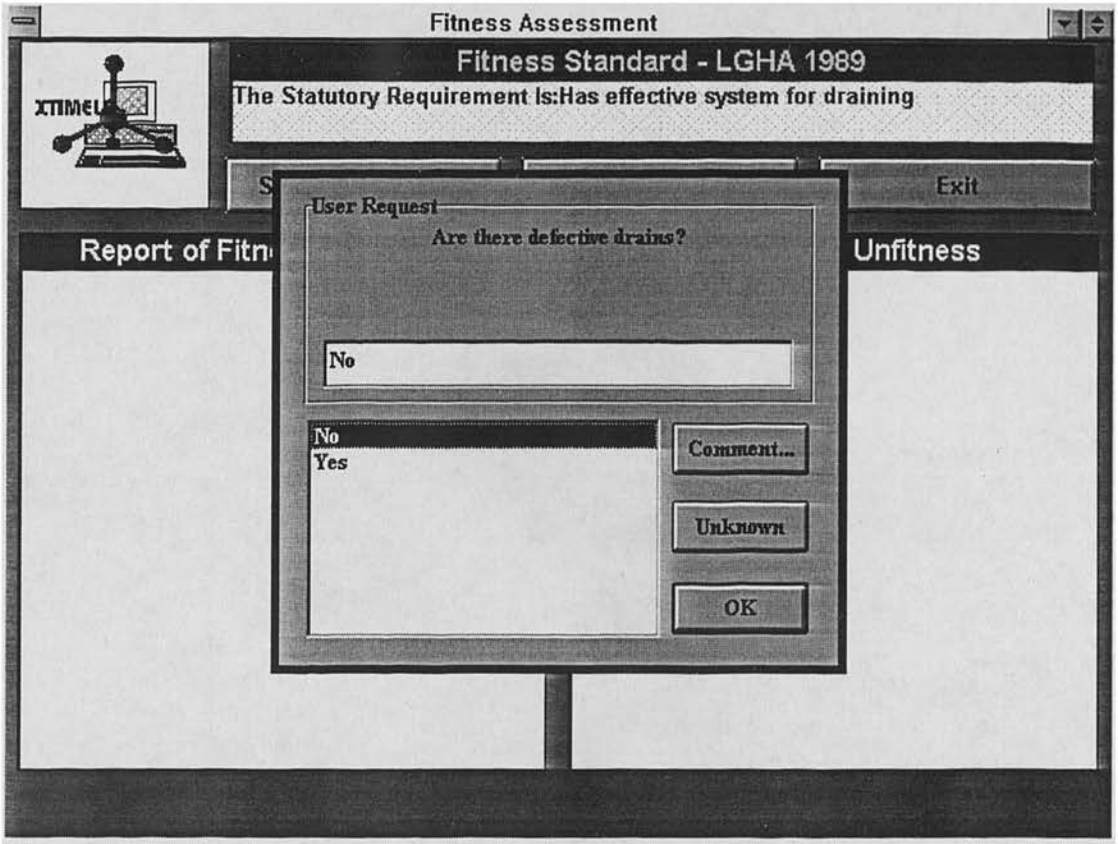
Screen 152: Input information about the condition of the building components



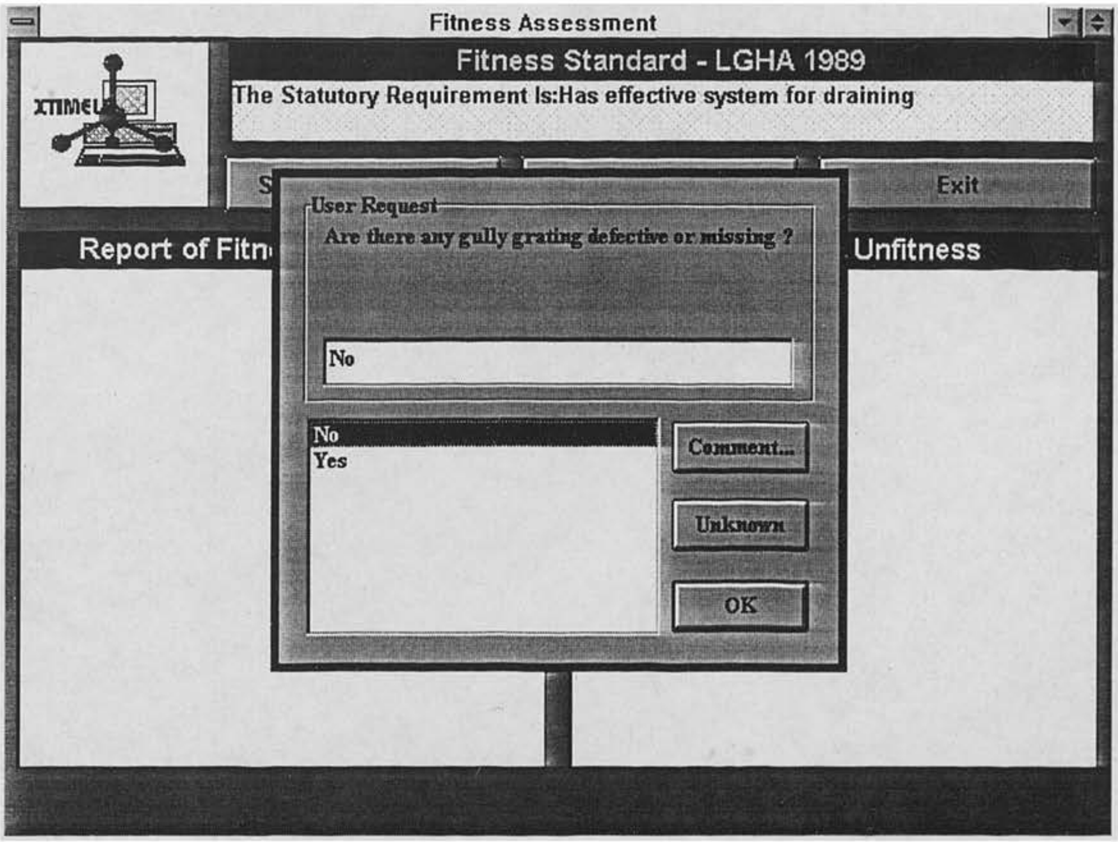
Screen 153: Input information about the condition of the building components



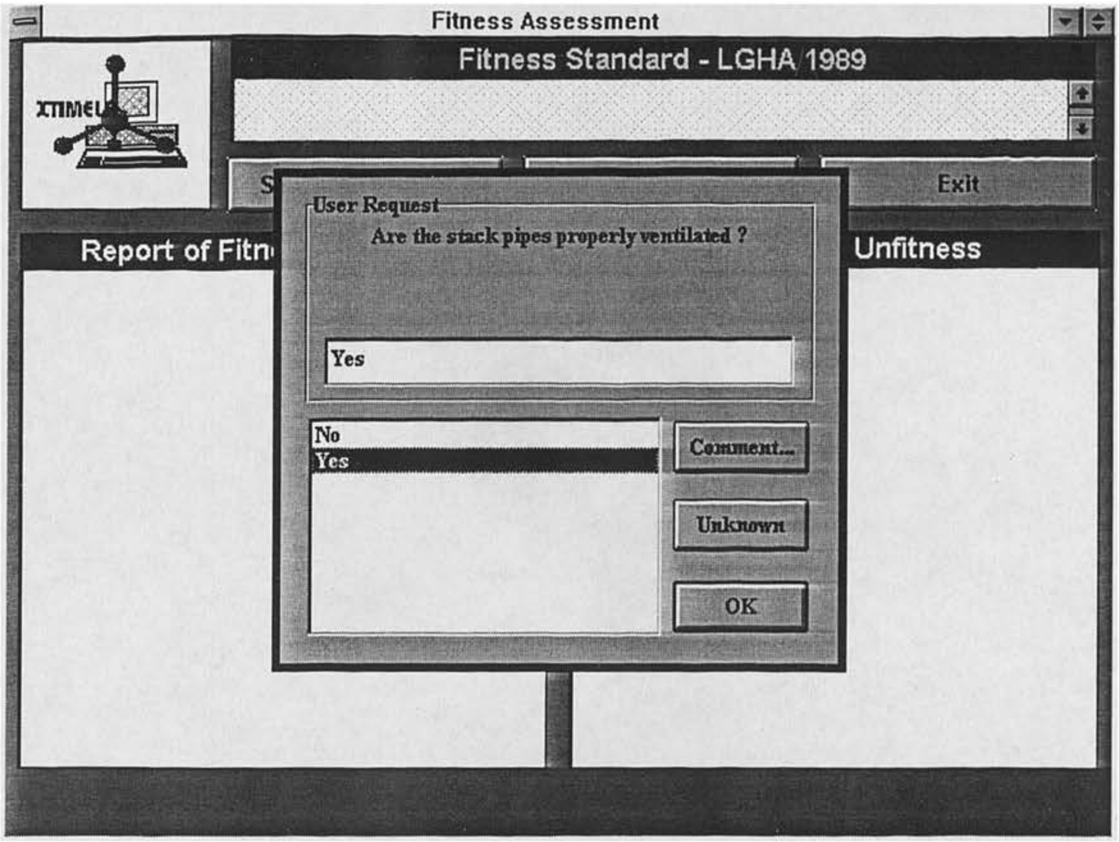
Screen 154: Input information about the condition of the building components



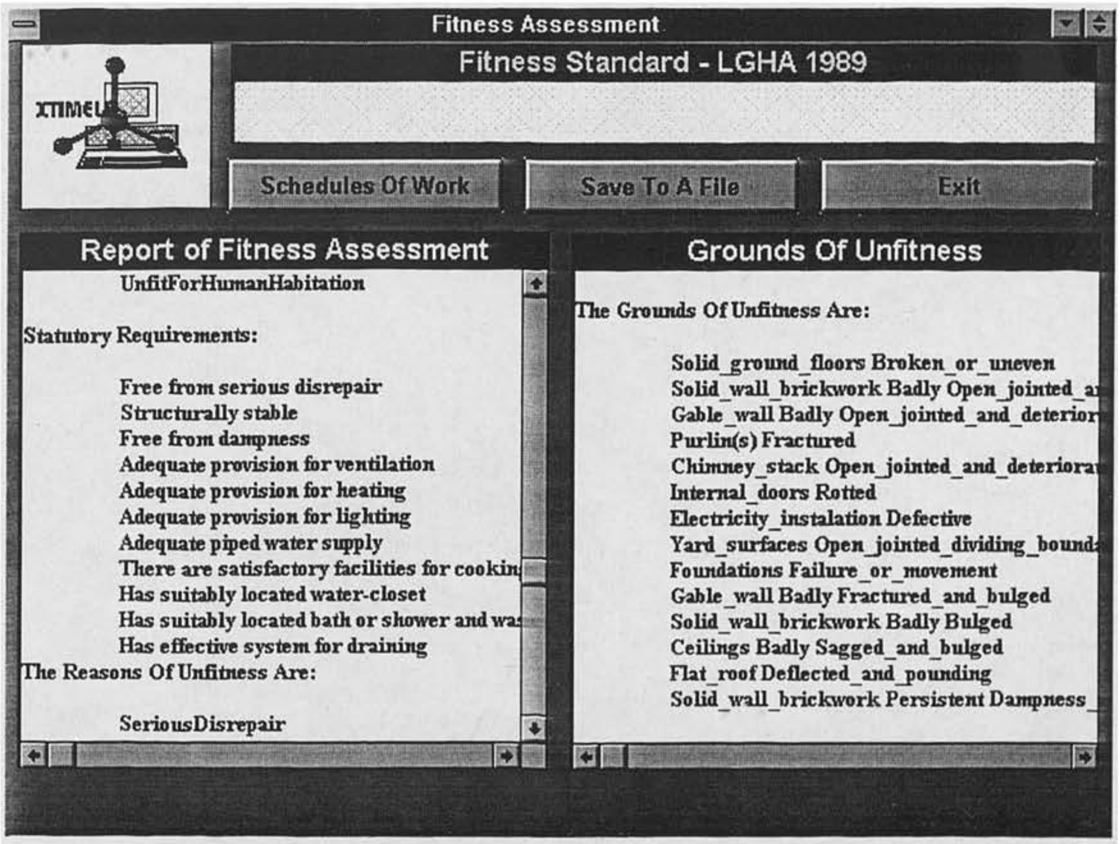
Screen 155: Input information about the condition of the building components



Screen 156: Input information about the condition of the building components

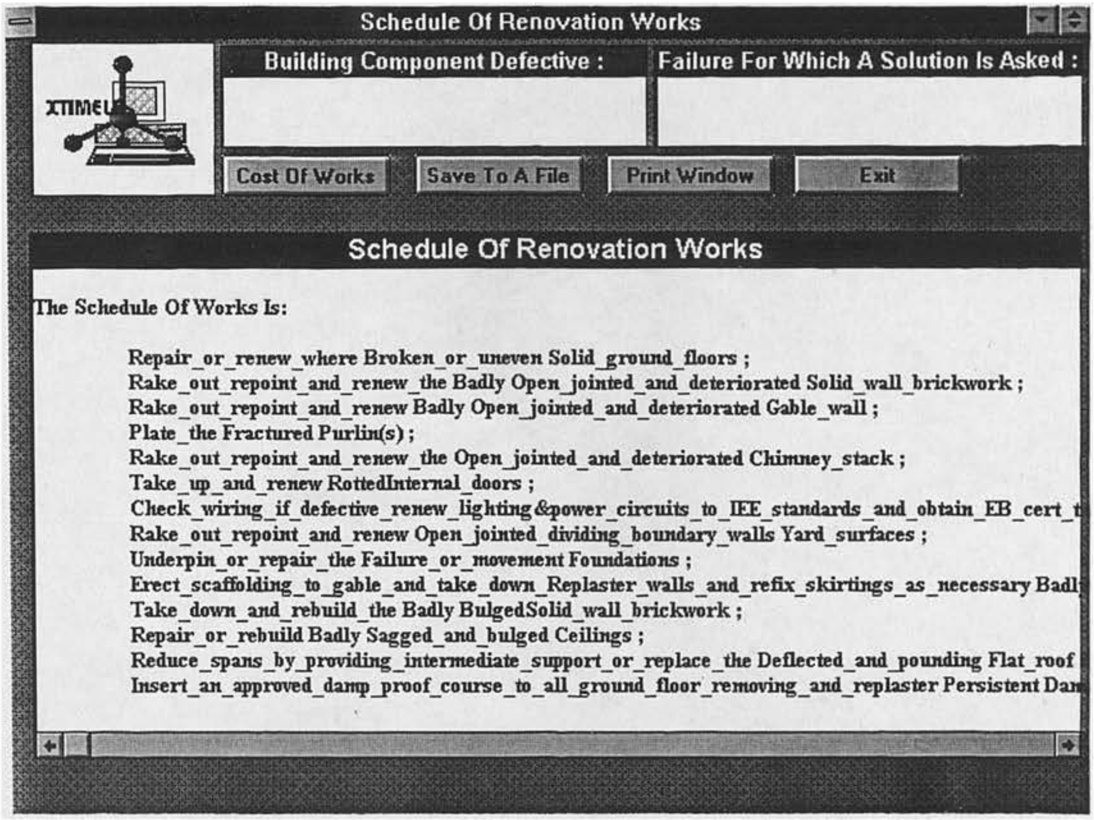


Screen 157: Input information about the condition of the building components



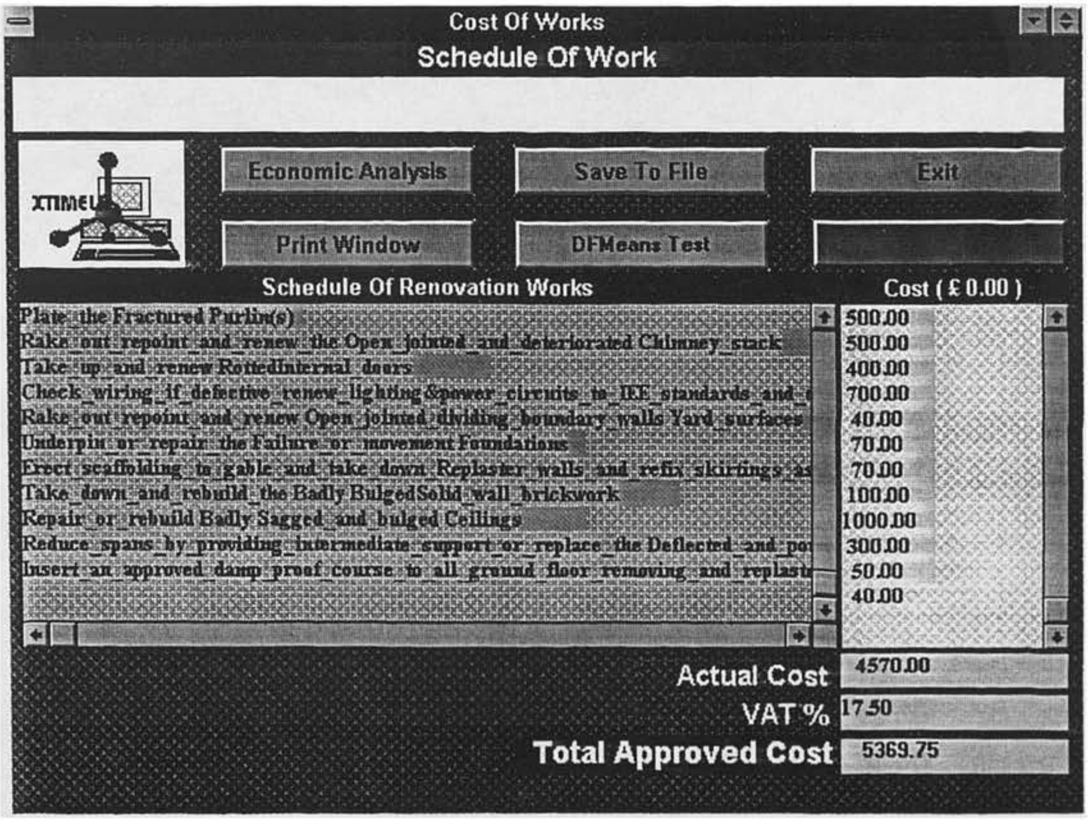
Screen 158: Partial view of the fitness assessment report

10.5- SCHEDULES OF WORK



Screen 159: Schedules of work retrived or adapted by the system

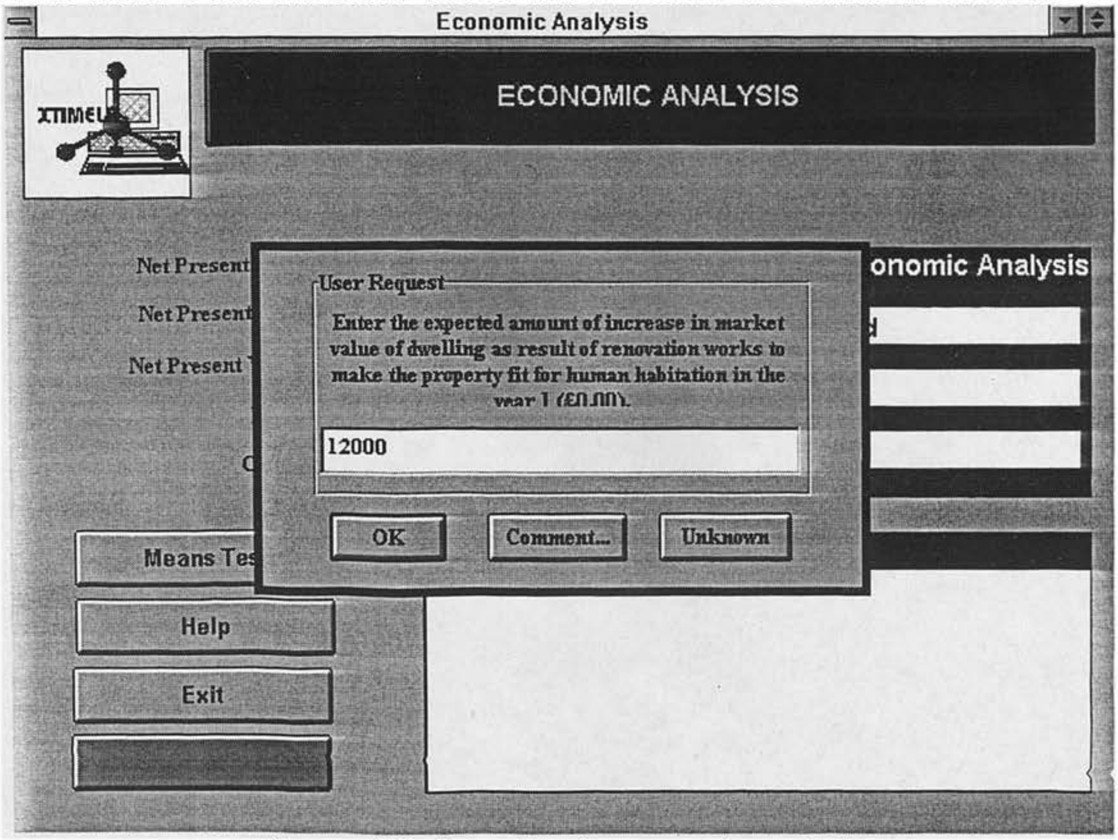
10.6- COST OF WORKS



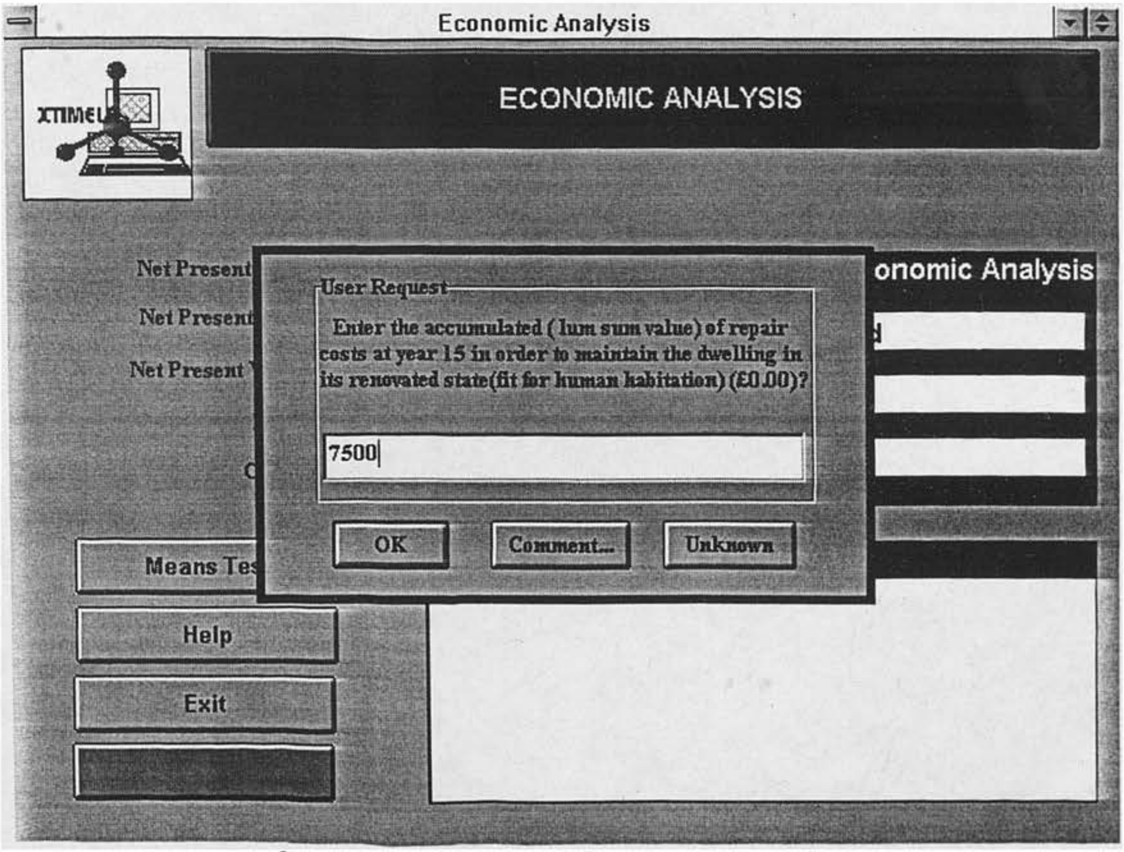
Screen 160: Cost of renovation works

10.7- ECONOMIC ANALYSIS

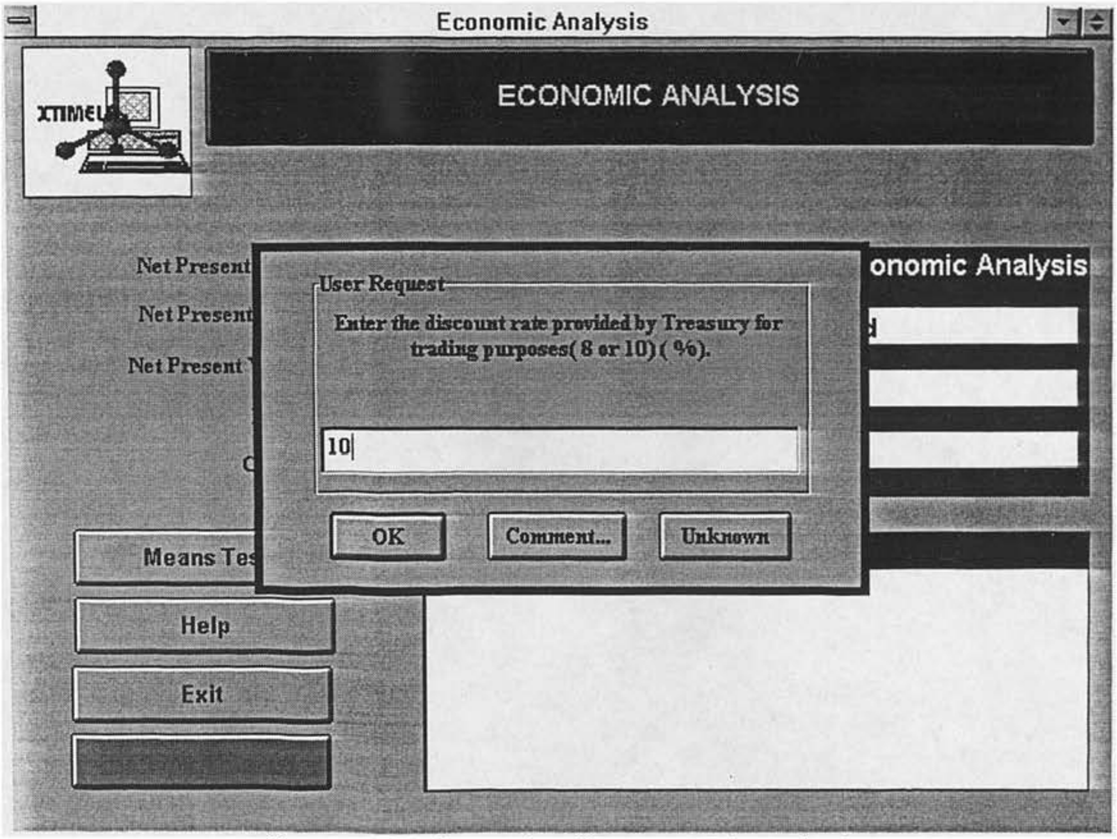
CALCULATING THE NET PRESENT VALUE OF RENOVATION WORKS



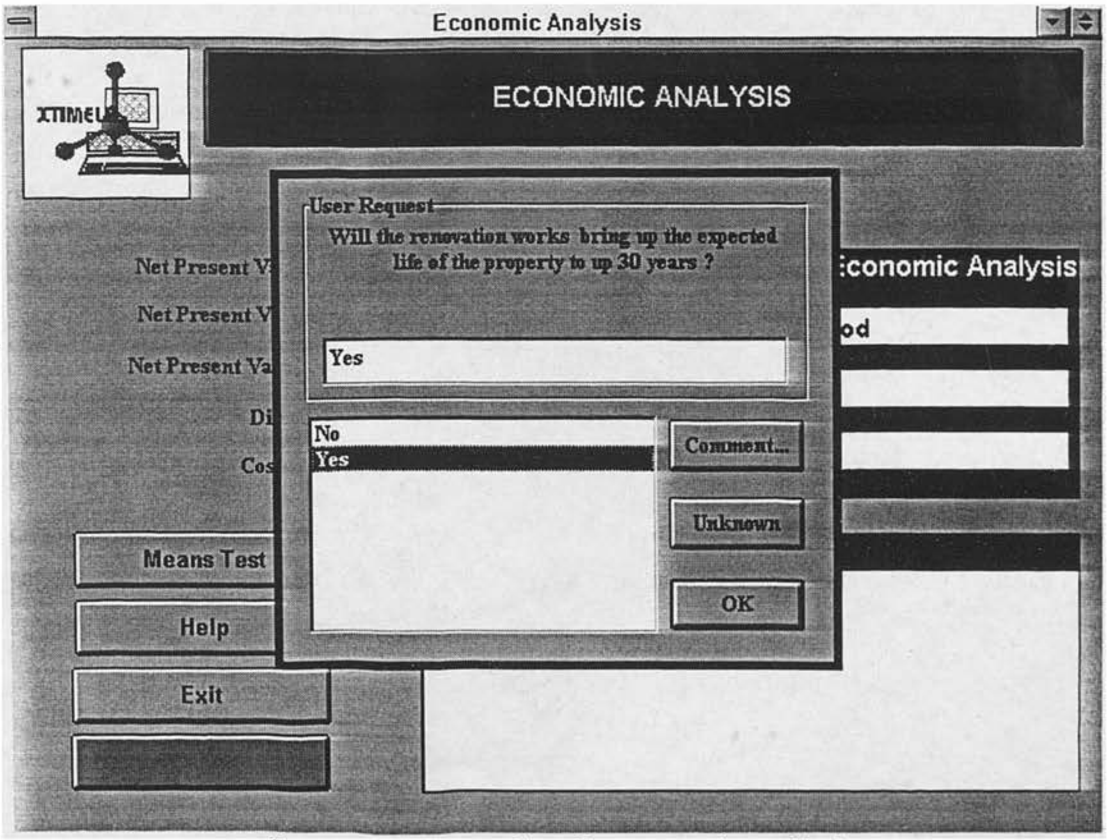
Screen 161: Input data for economic analysis



Screen 162: Input data for economic analysis

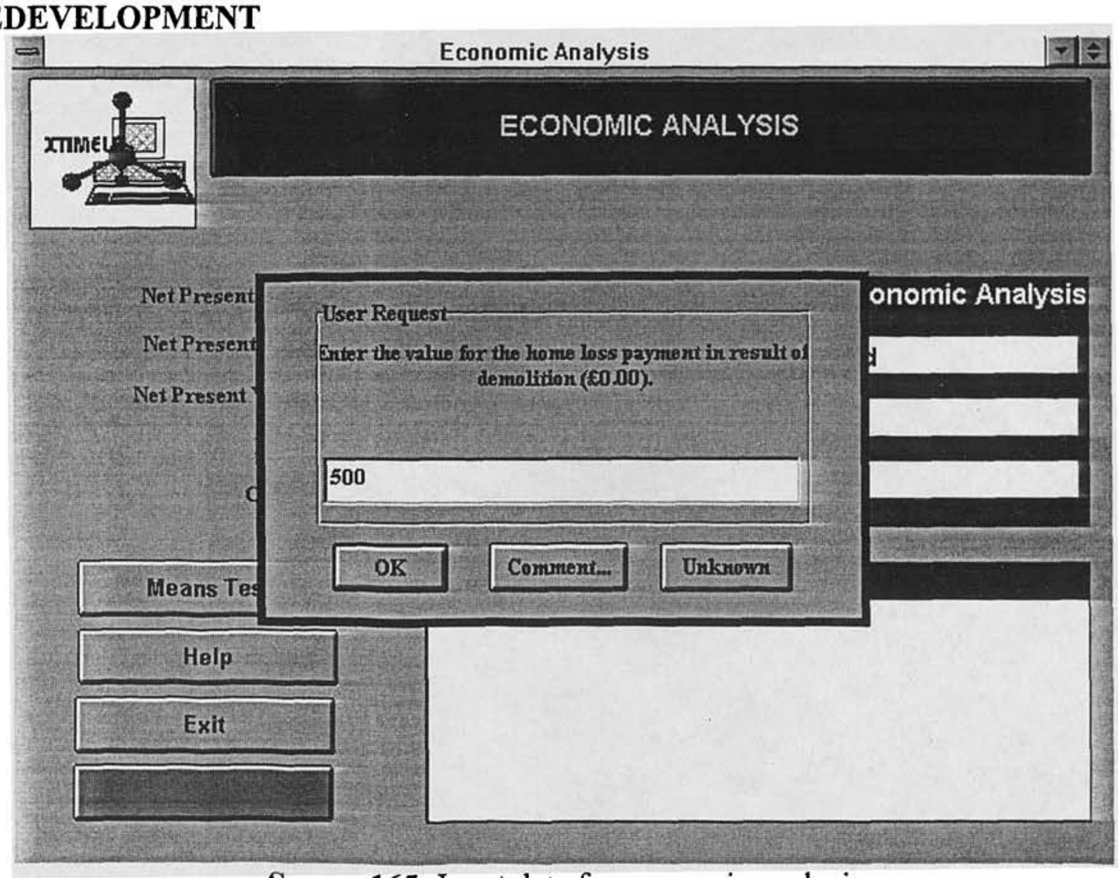


Screen 163: Input data for economic analysis

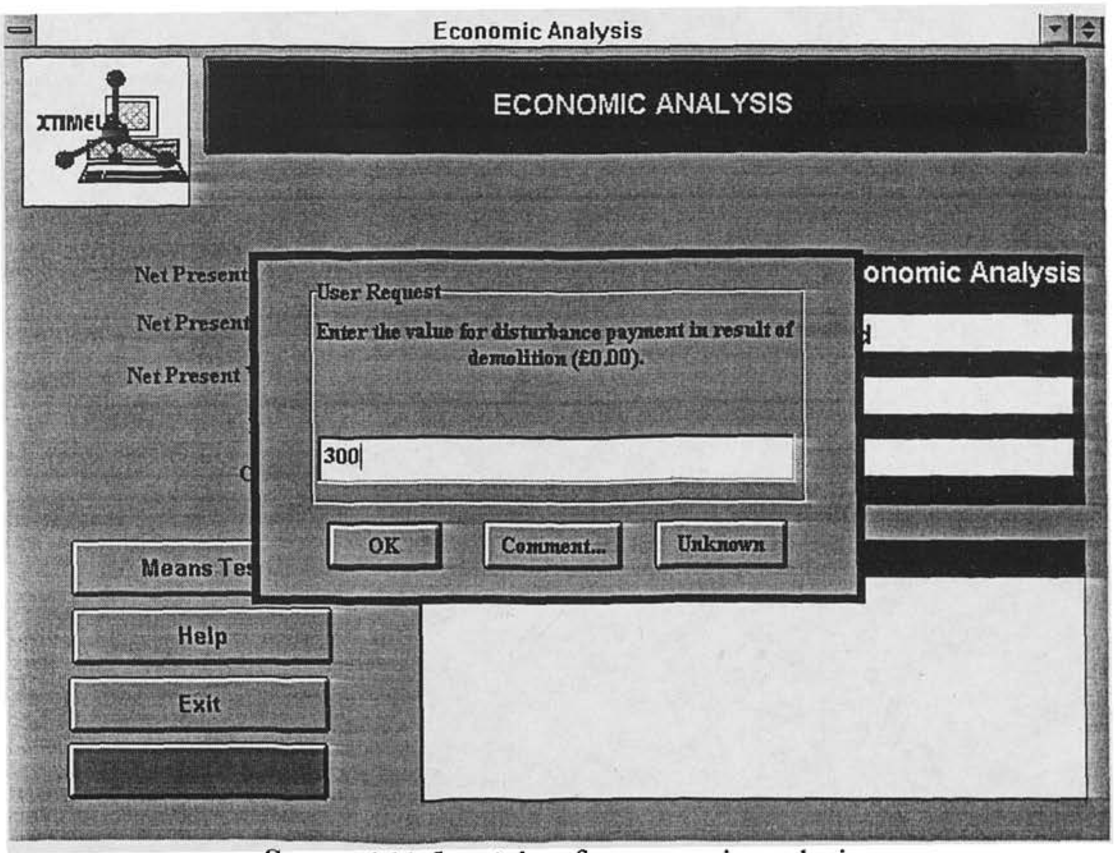


Screen 164: Input data for economic analysis

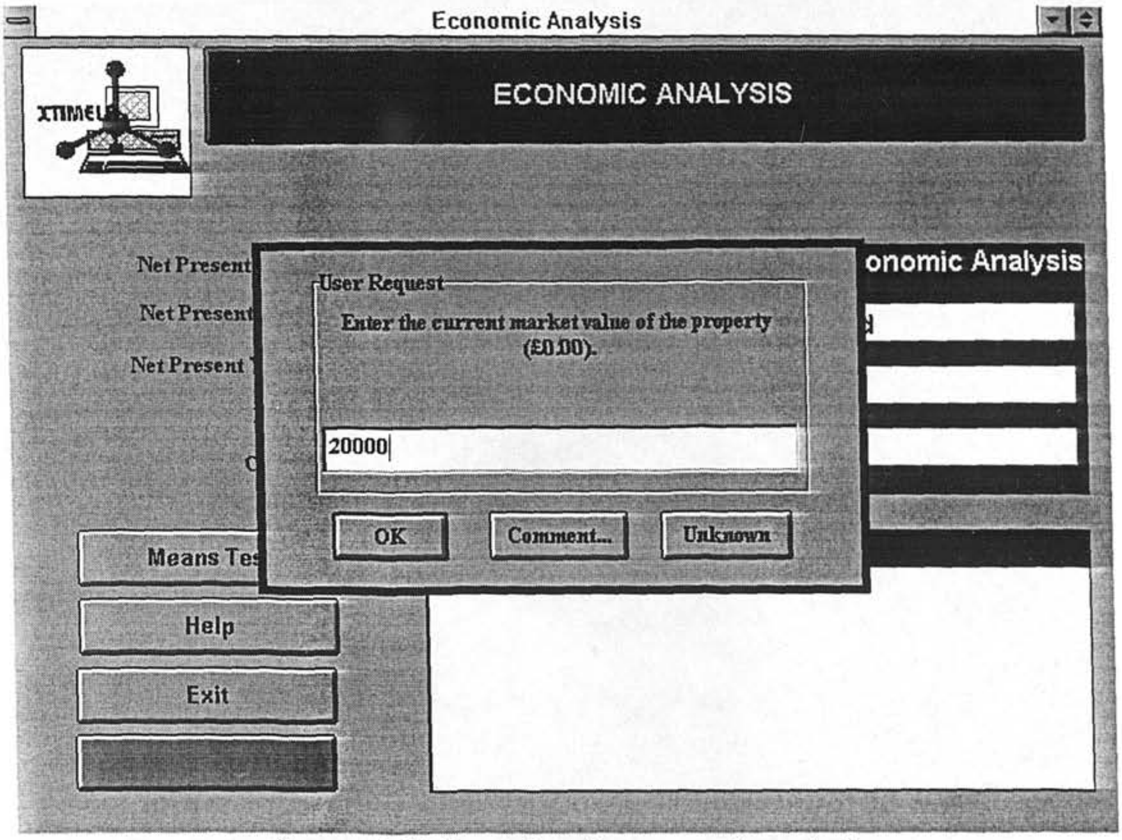
CALCULATING THE NET PRESENT VALUE OF DEMOLITON WITHOUT REDEVELOPMENT



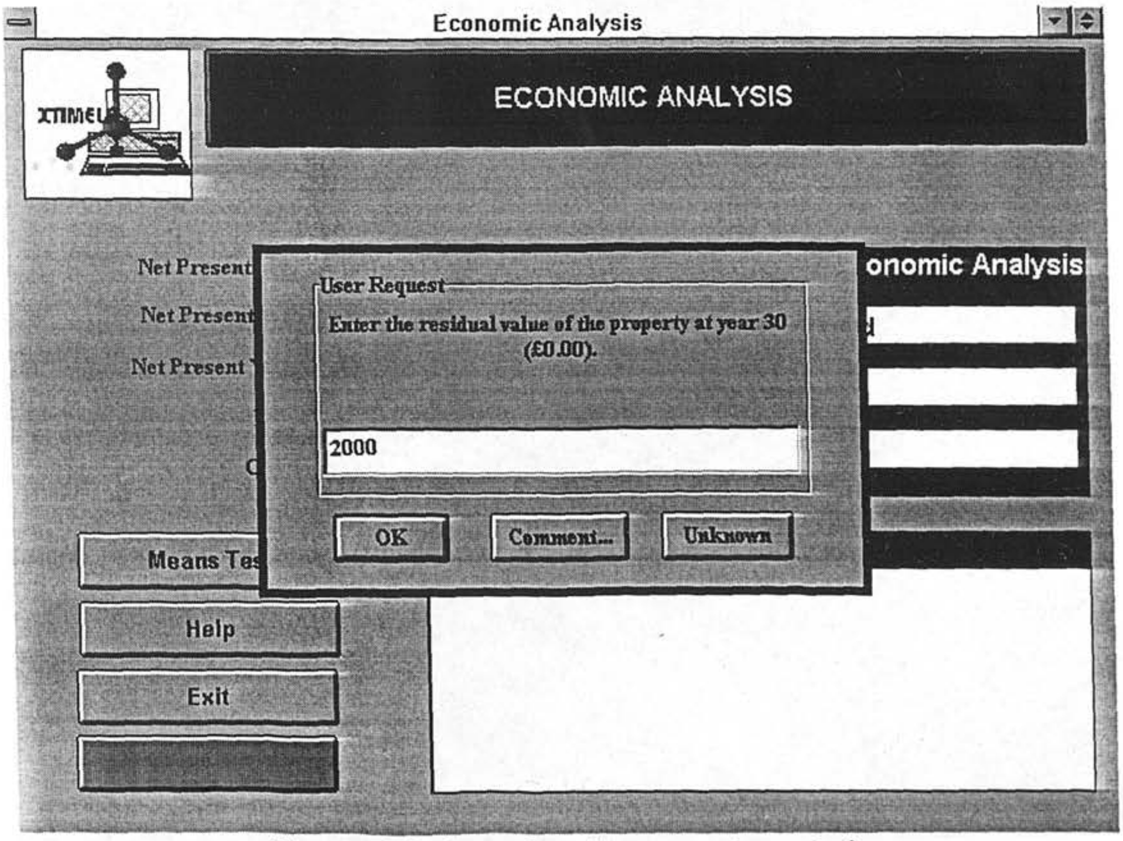
Screen 165: Input data for economic analysis



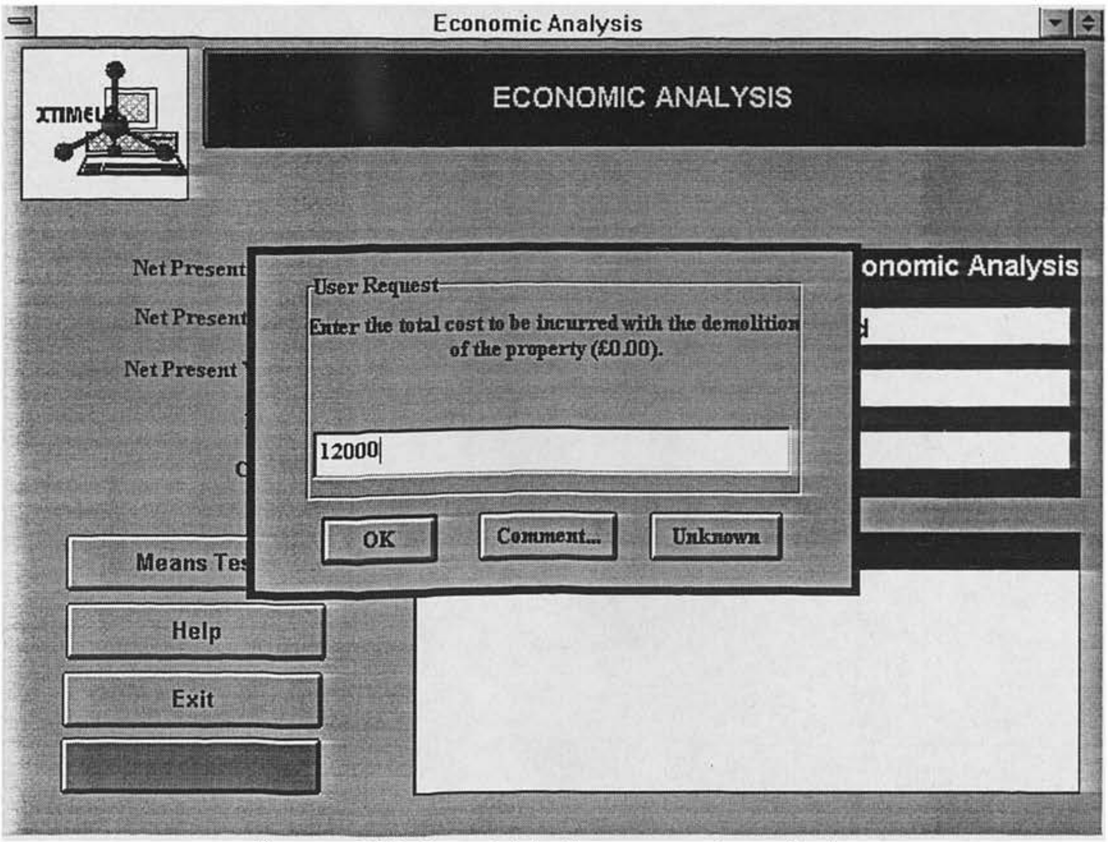
Screen 166: Input data for economic analysis



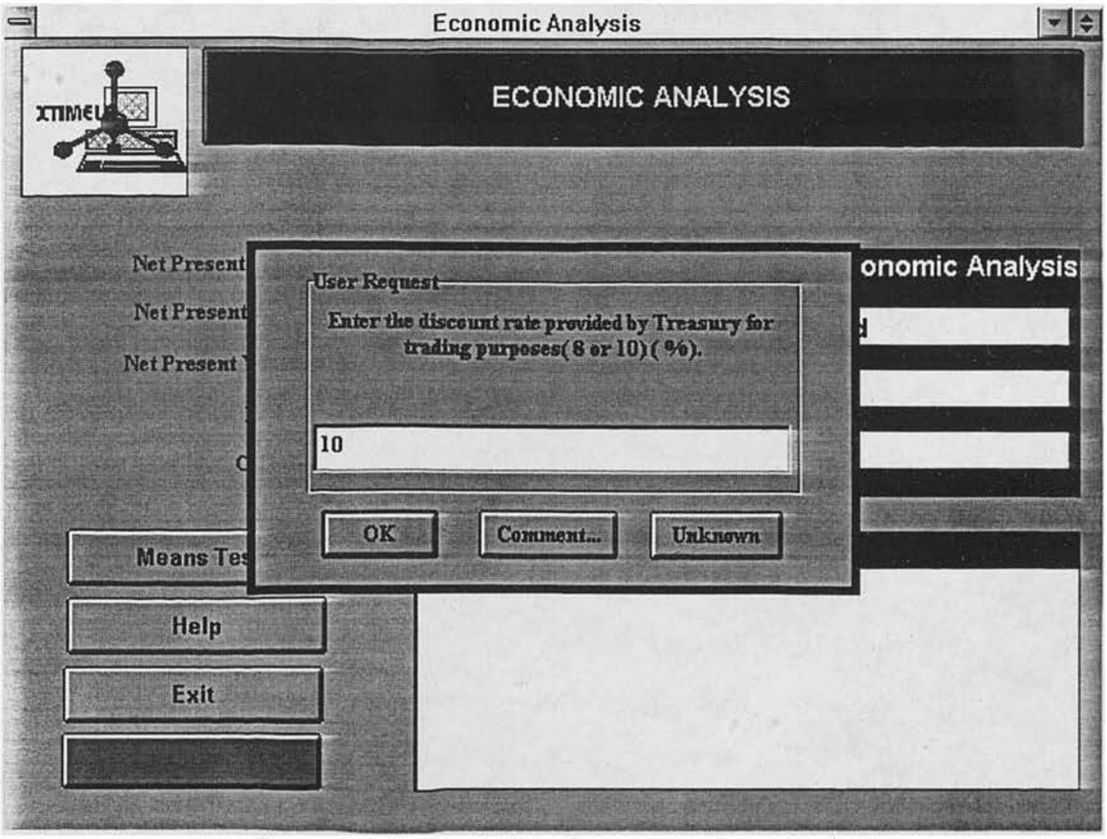
Screen 167: Input data for economic analysis



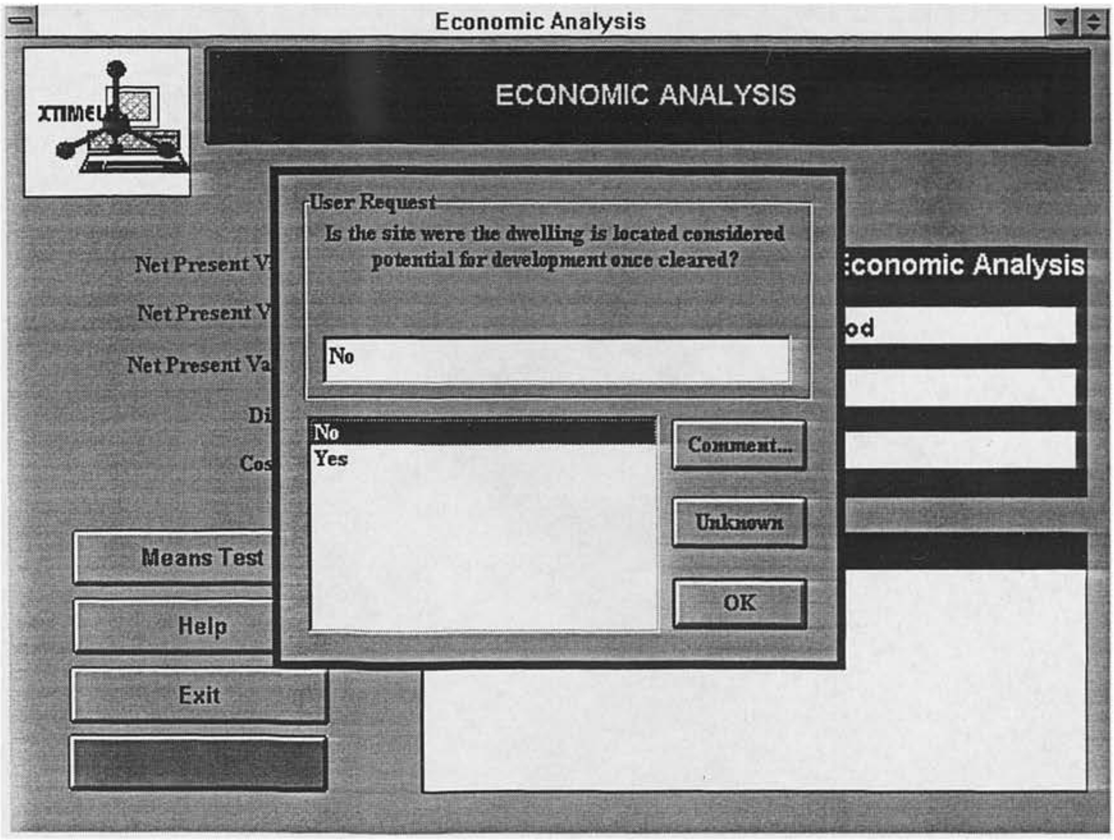
Screen 168: Input data for economic analysis



Screen 169: Input data for economic analysis

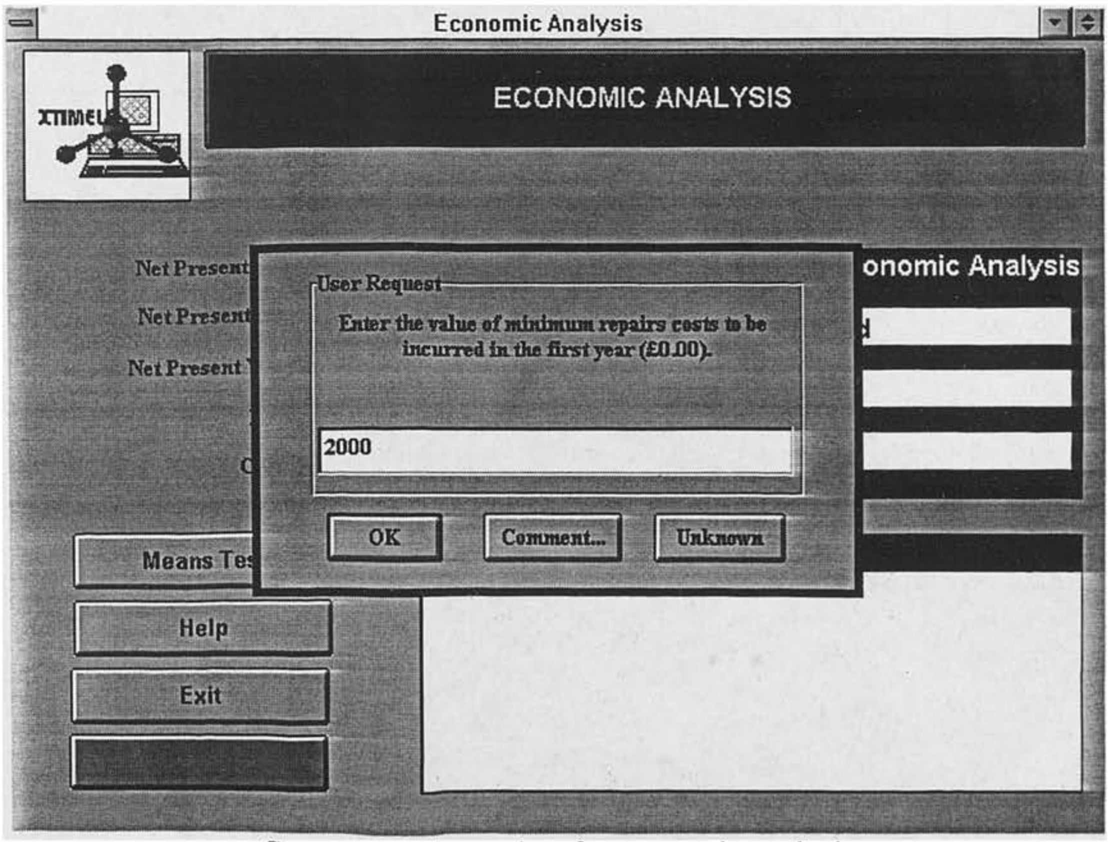


Screen 170: Input data for economic analysis

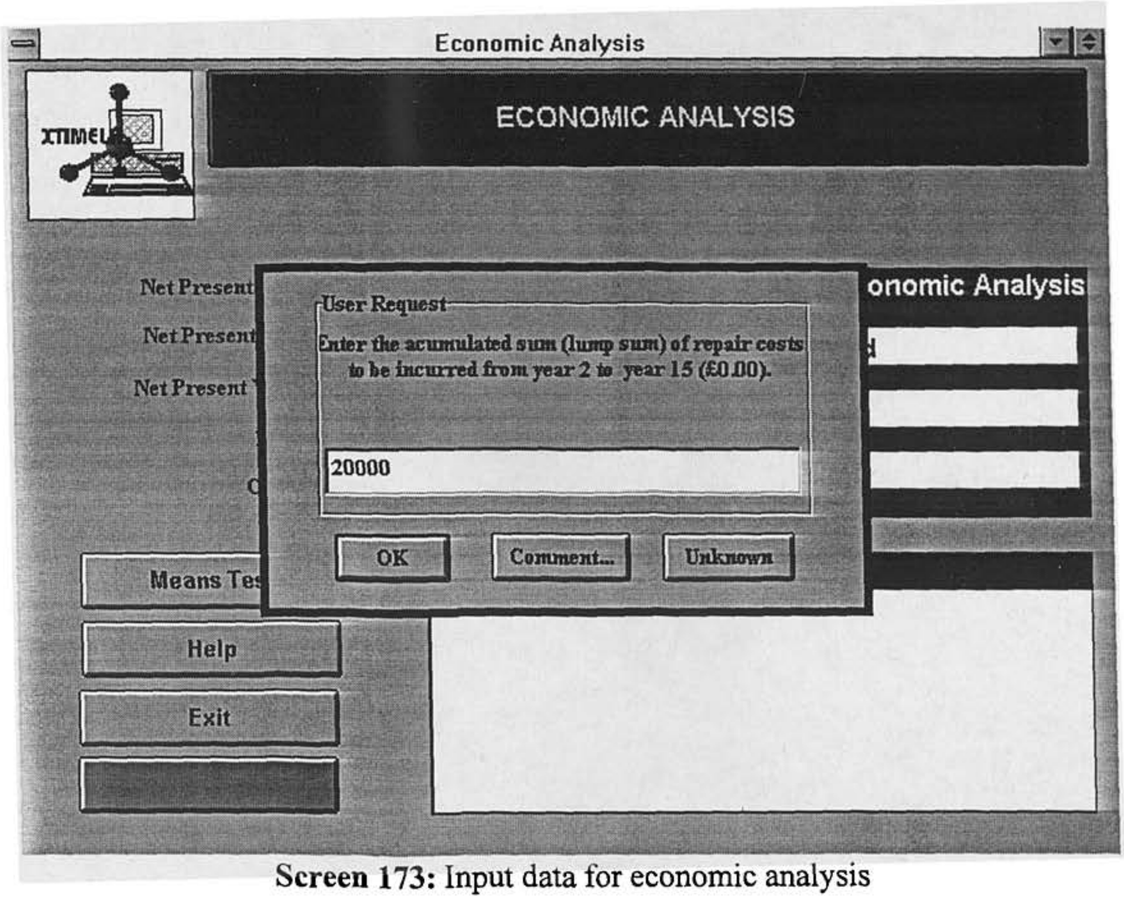


Screen 171: Input data for economic analysis

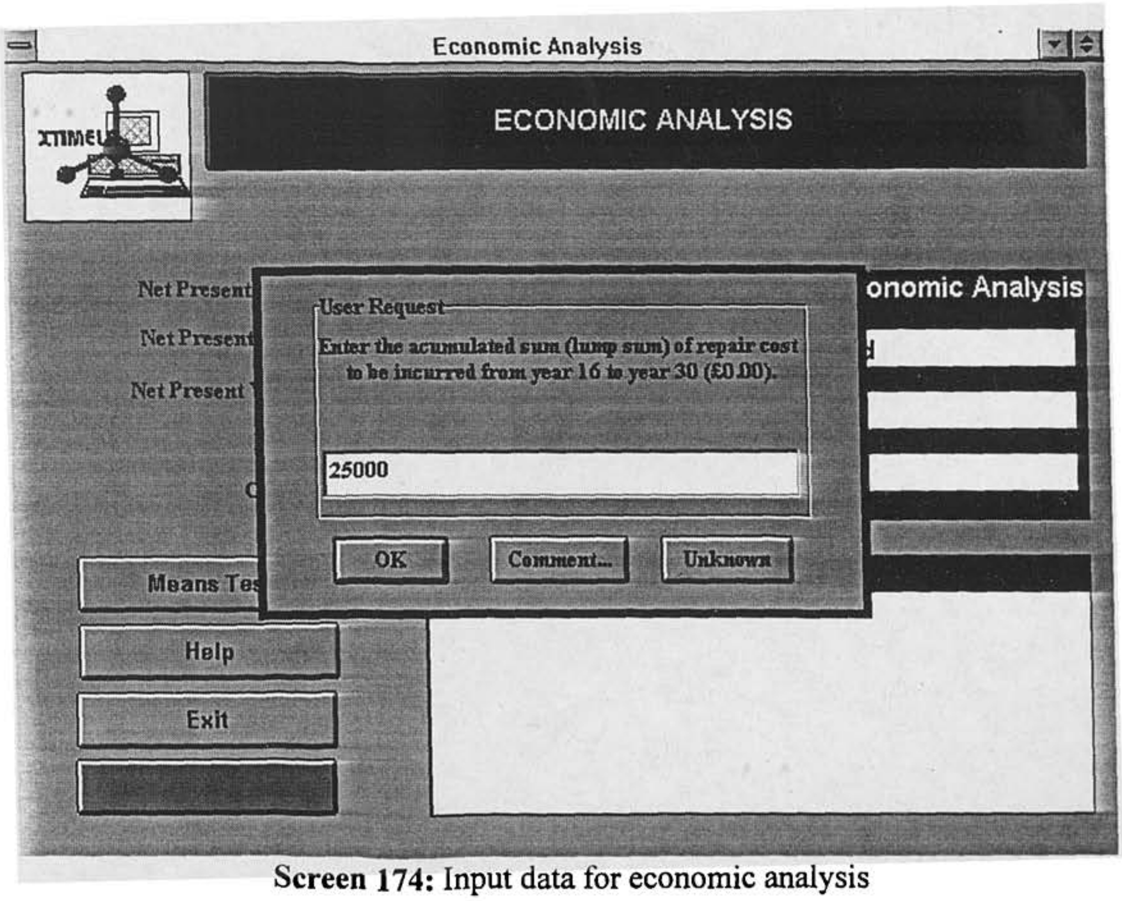
CALCULATING THE NET PRESENT VALUE FOR MAINTAINING THE CURRENT STATE OF THE PROPERTY



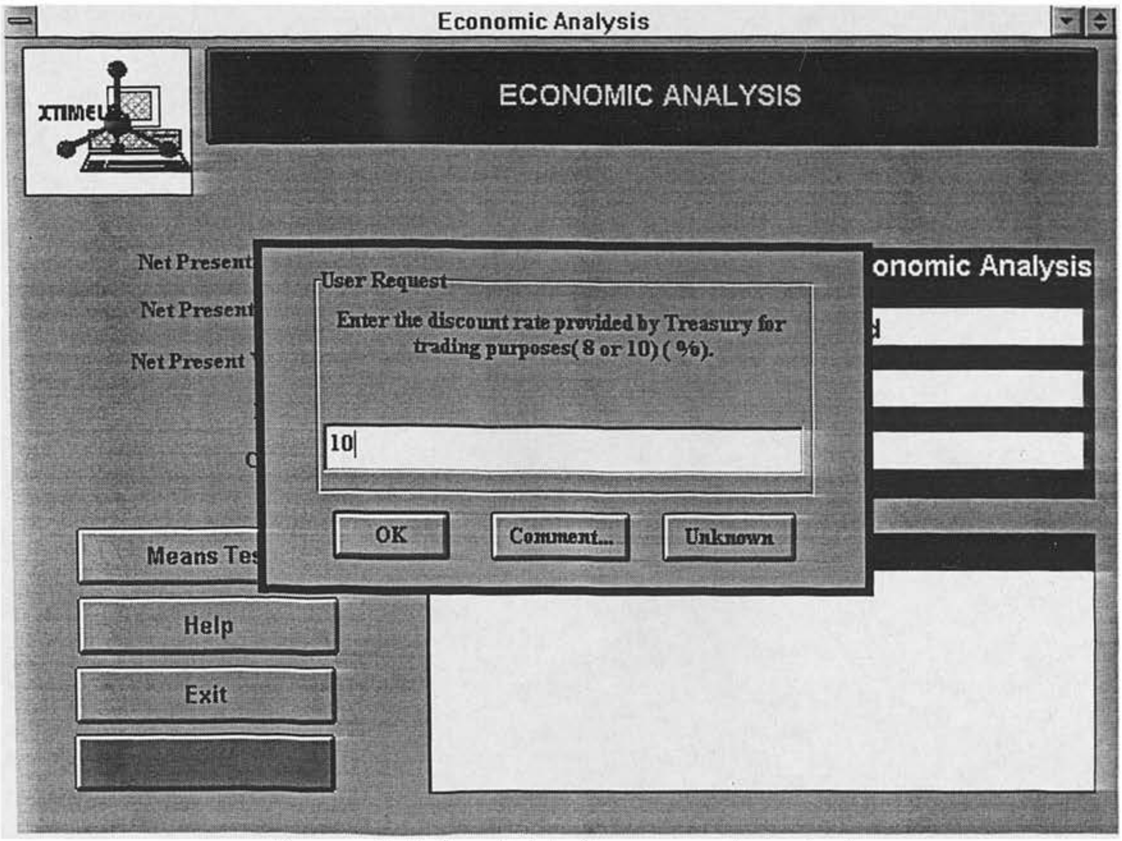
Screen 172: Input data for economic analysis



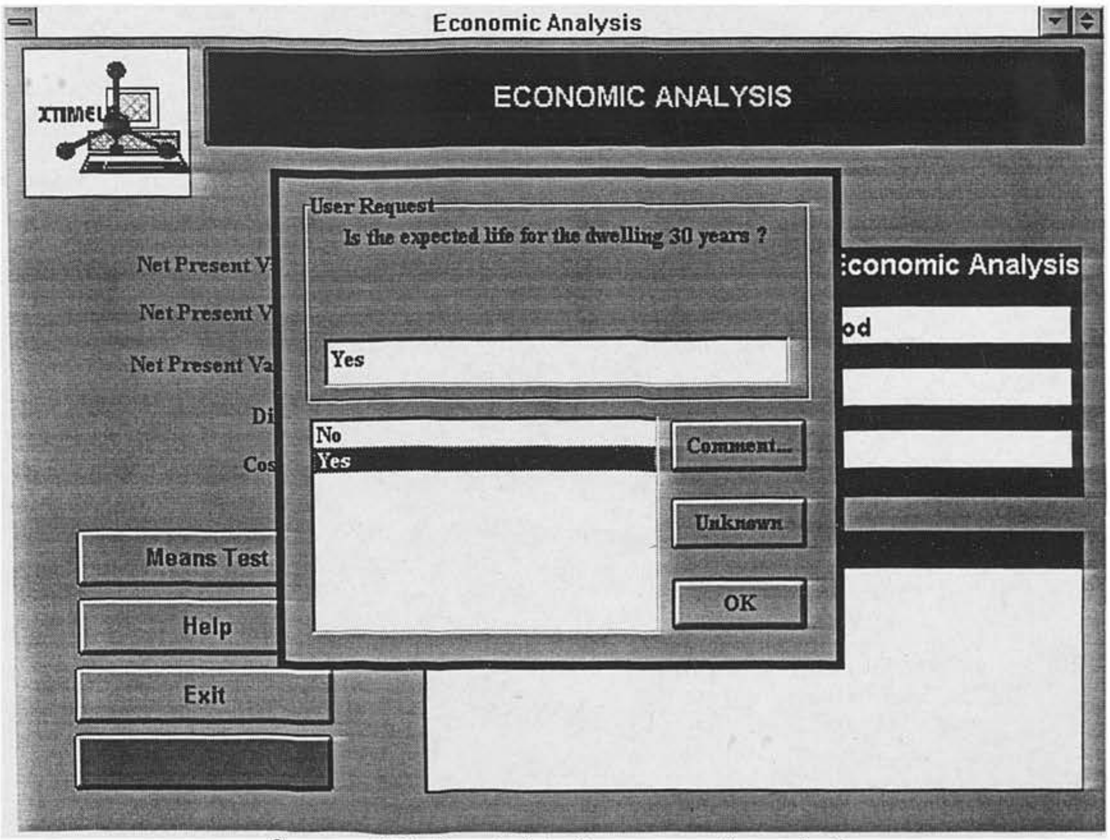
Screen 173: Input data for economic analysis



Screen 174: Input data for economic analysis

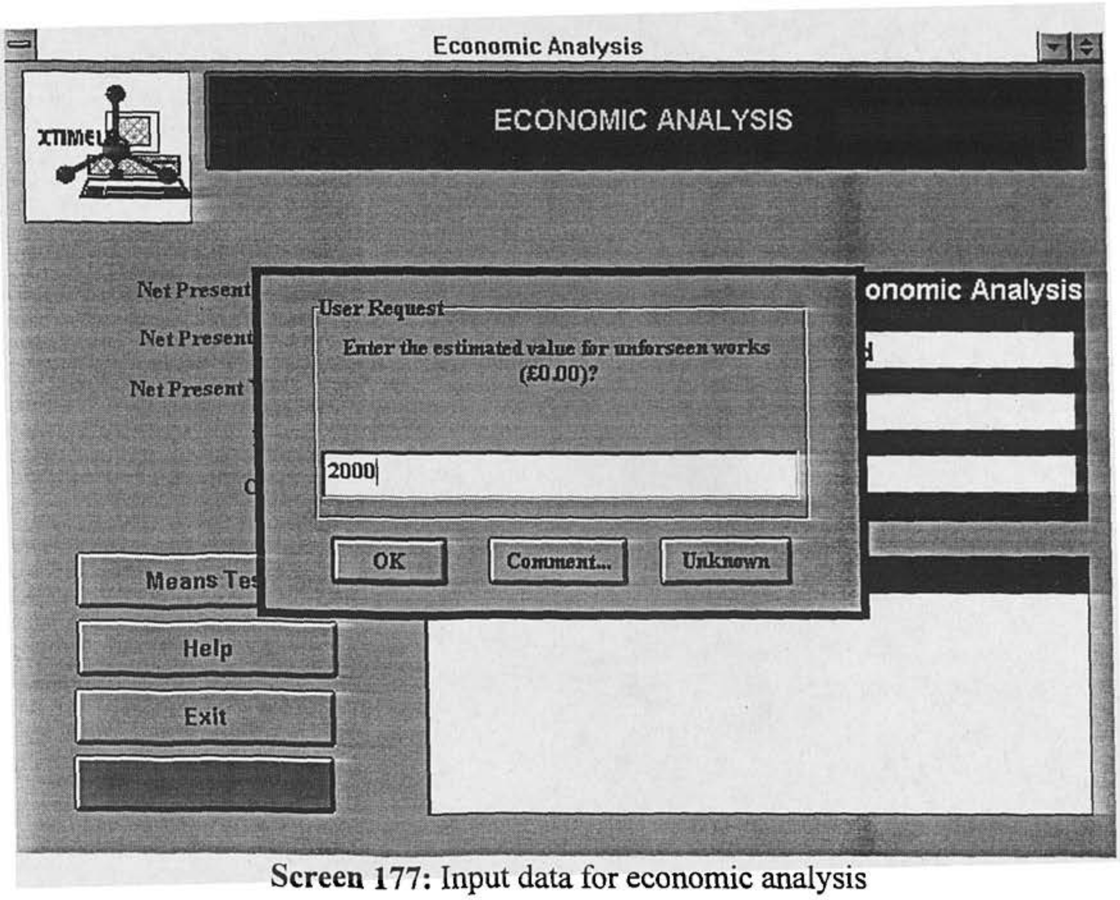


Screen 175: Input data for economic analysis

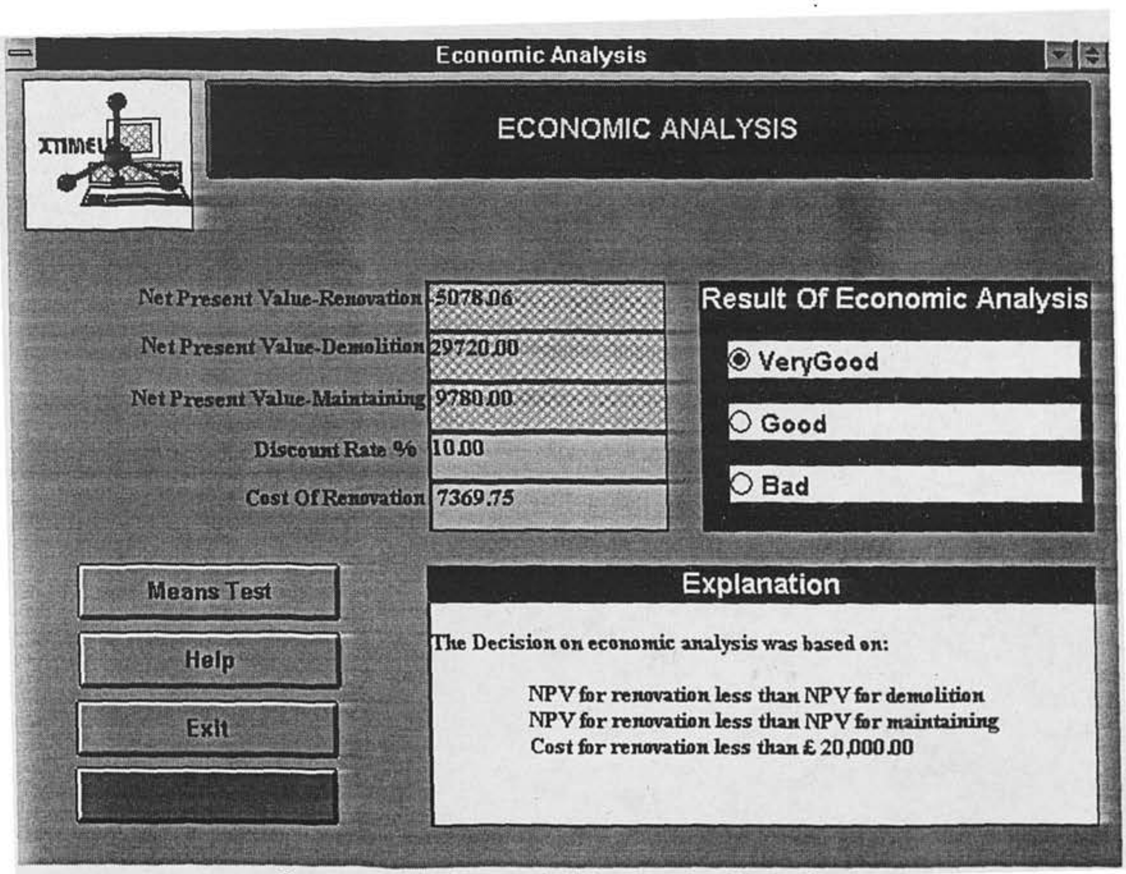


Screen 176: Input data for economic analysis

ASSESSING THE ECONOMIC MERITS



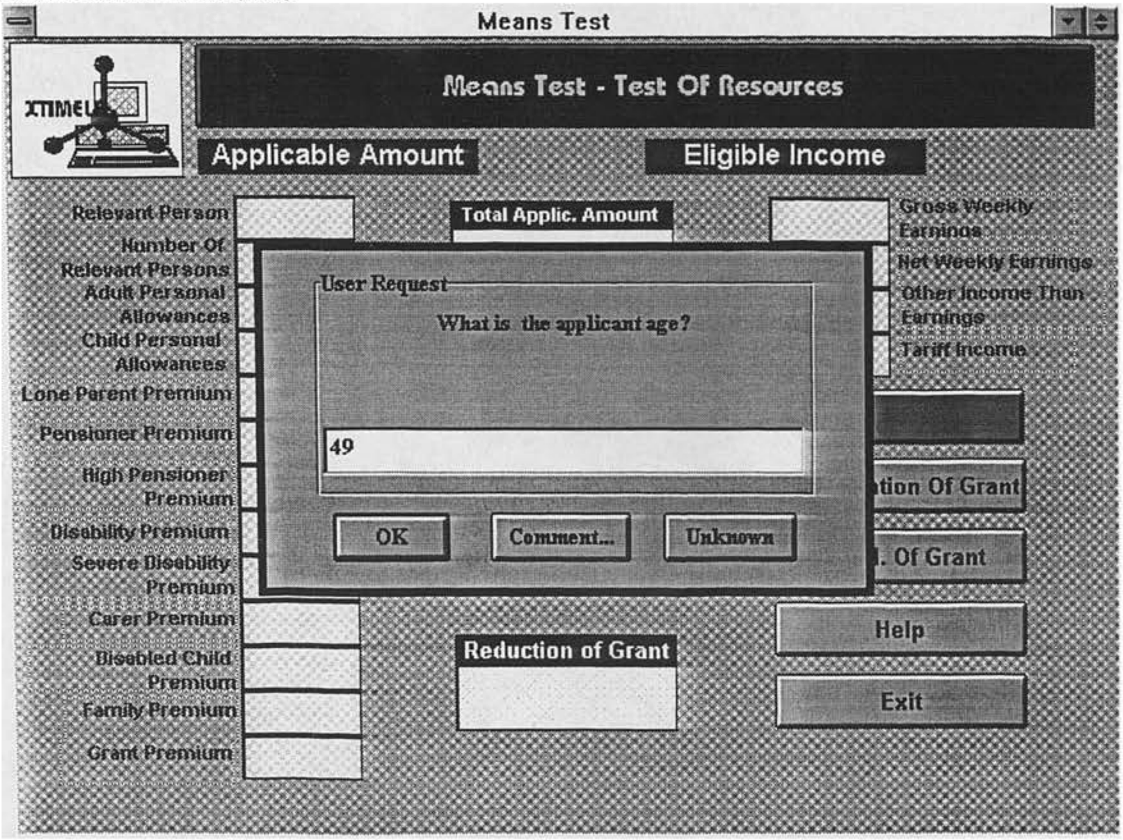
Screen 177: Input data for economic analysis



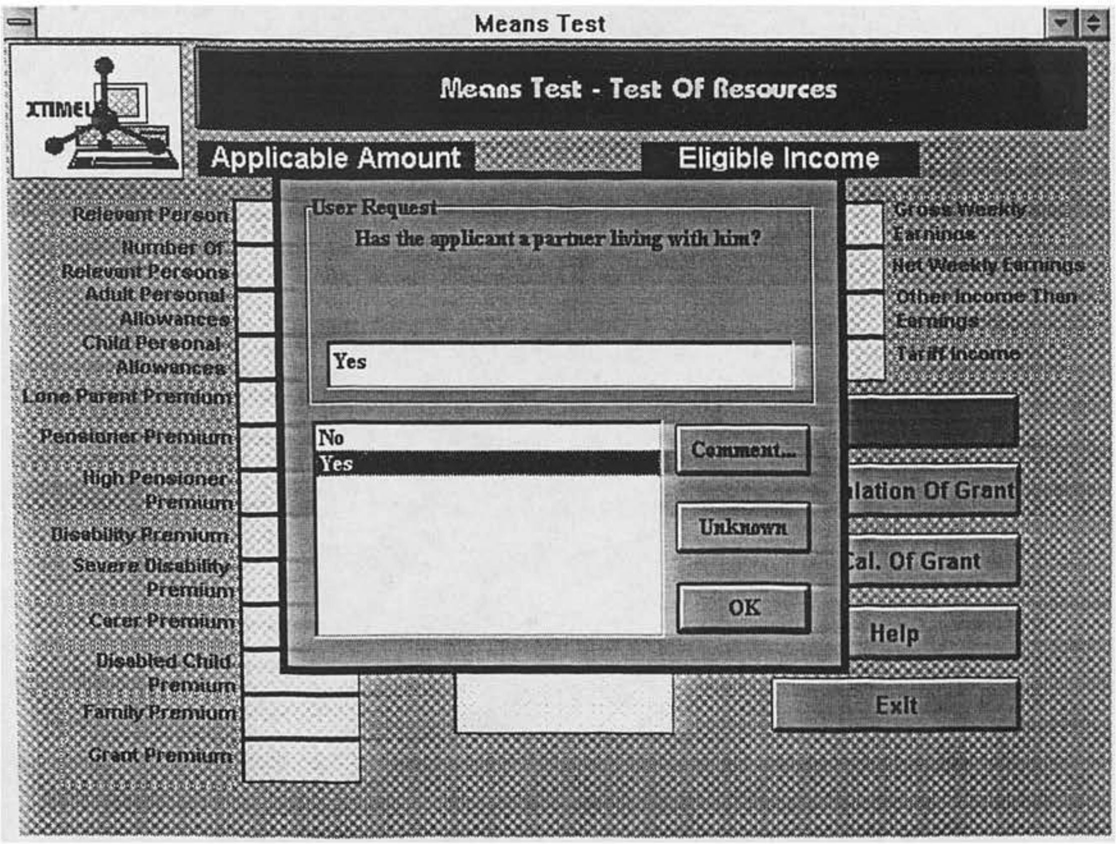
Screen 178: Report of the economic analysis

10.8- MEANS TEST

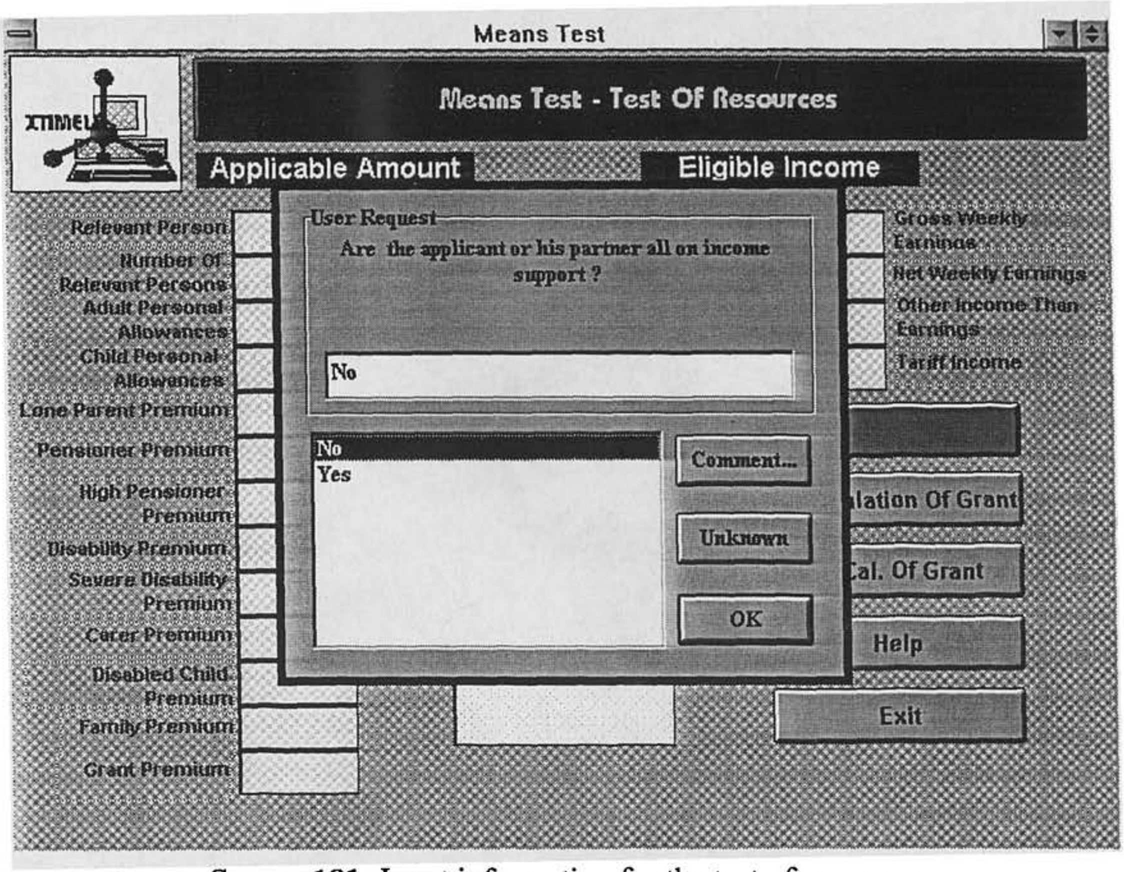
RELEVANT PERSONS



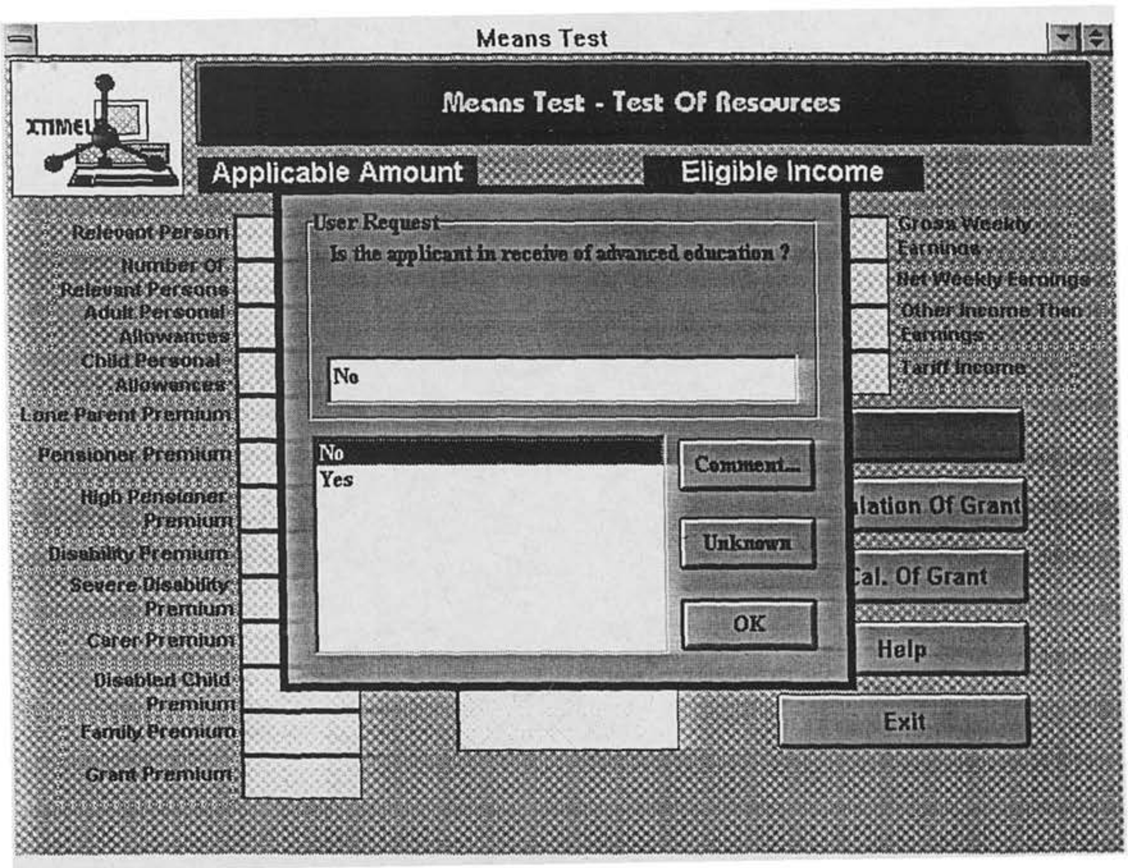
Screen 179: Input information for the test of resources



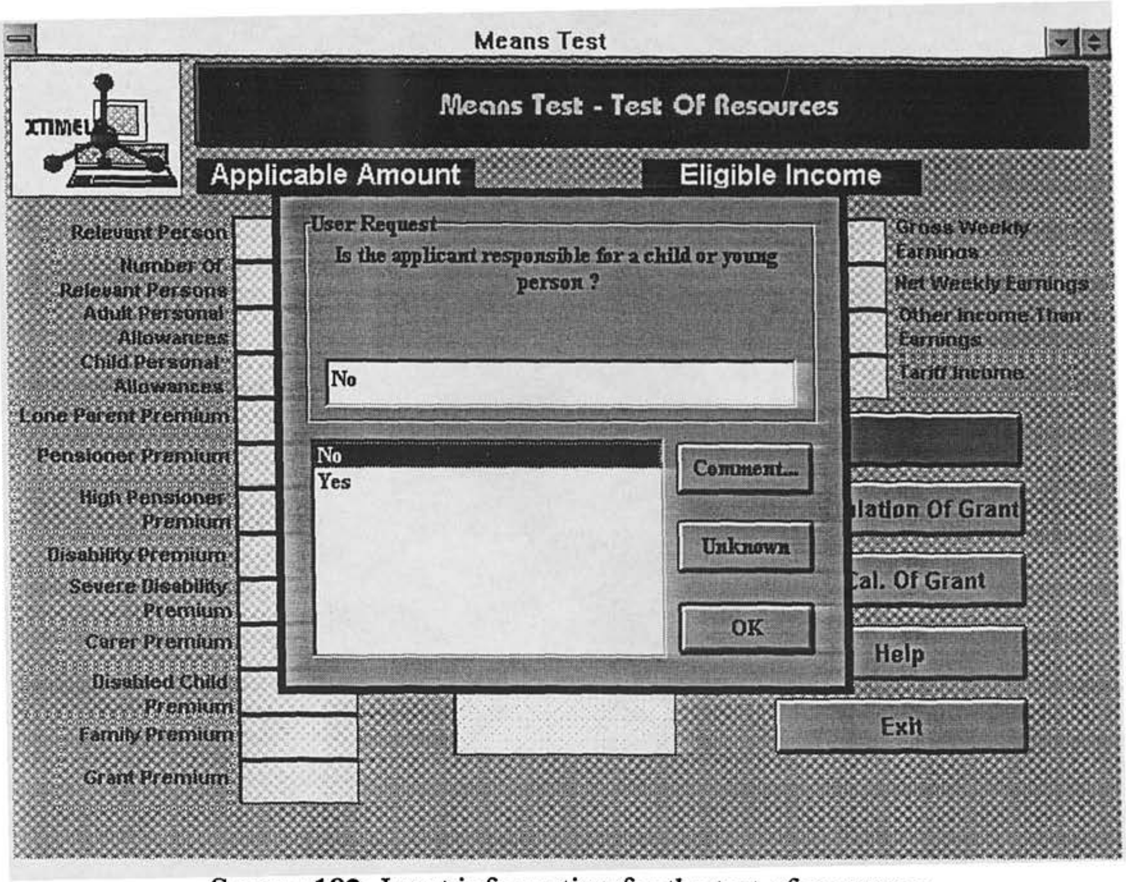
Screen 180: Input information for the test of resources



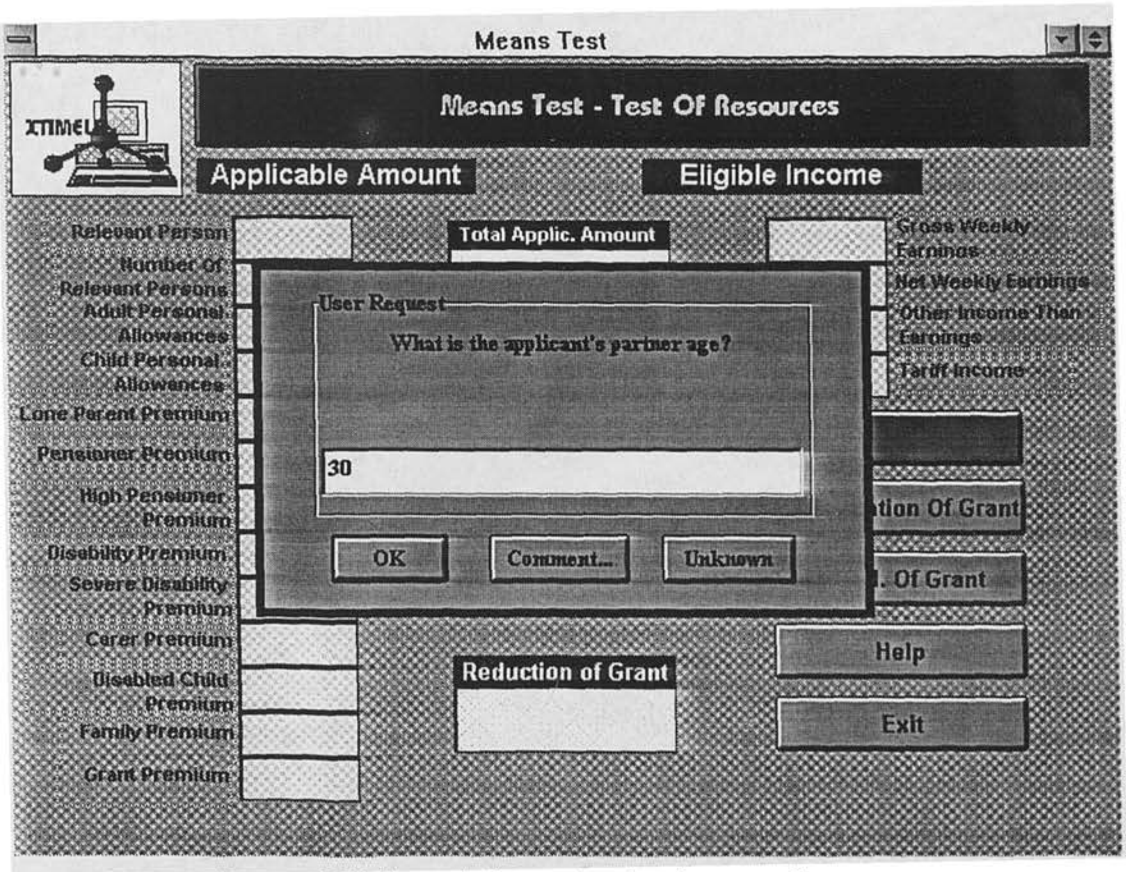
Screen 181: Input information for the test of resources



Screen 182: Input information for the test of resources

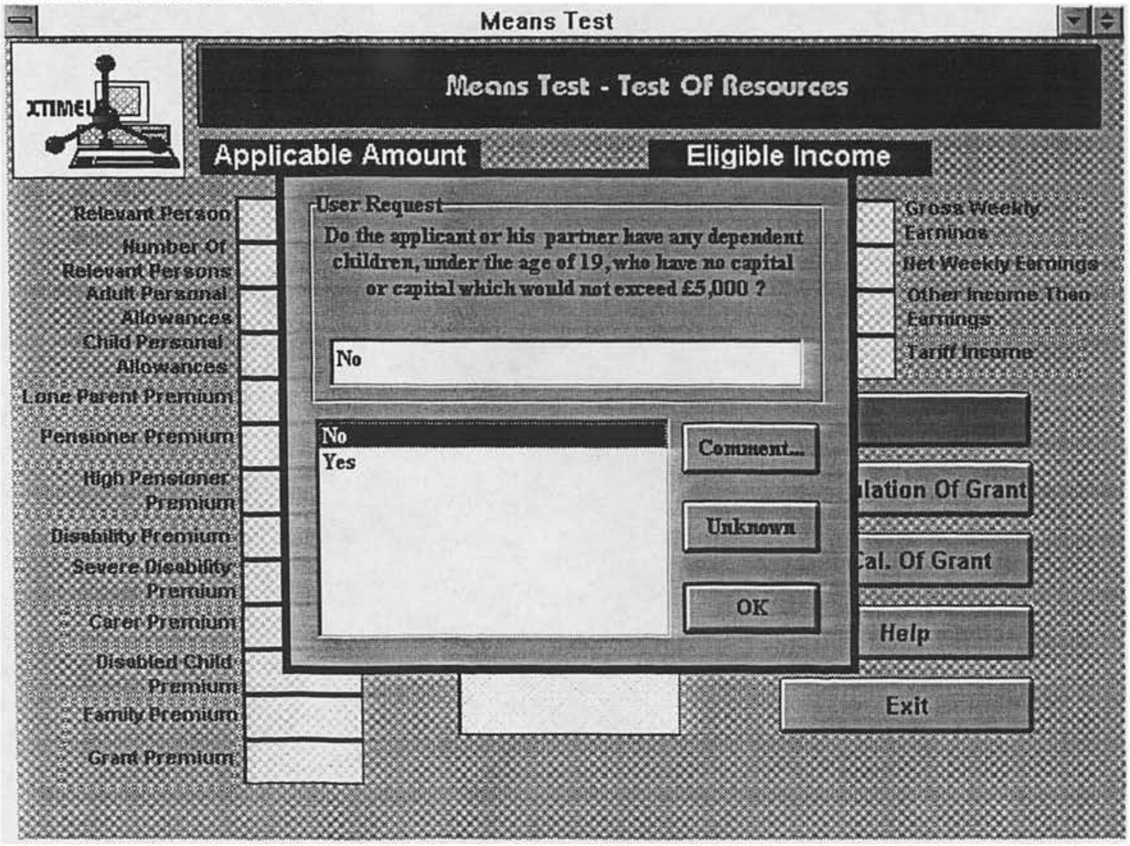


Screen 183: Input information for the test of resources

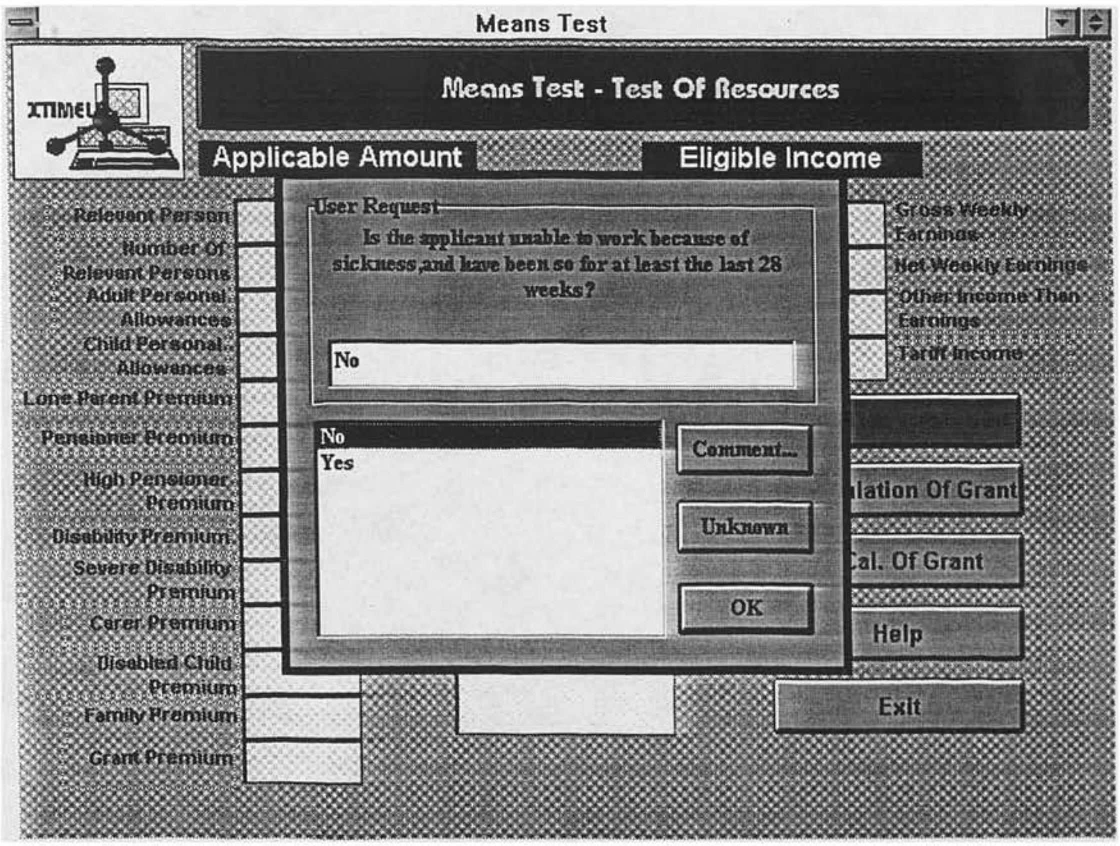


Screen 184: Input information for the test of resources

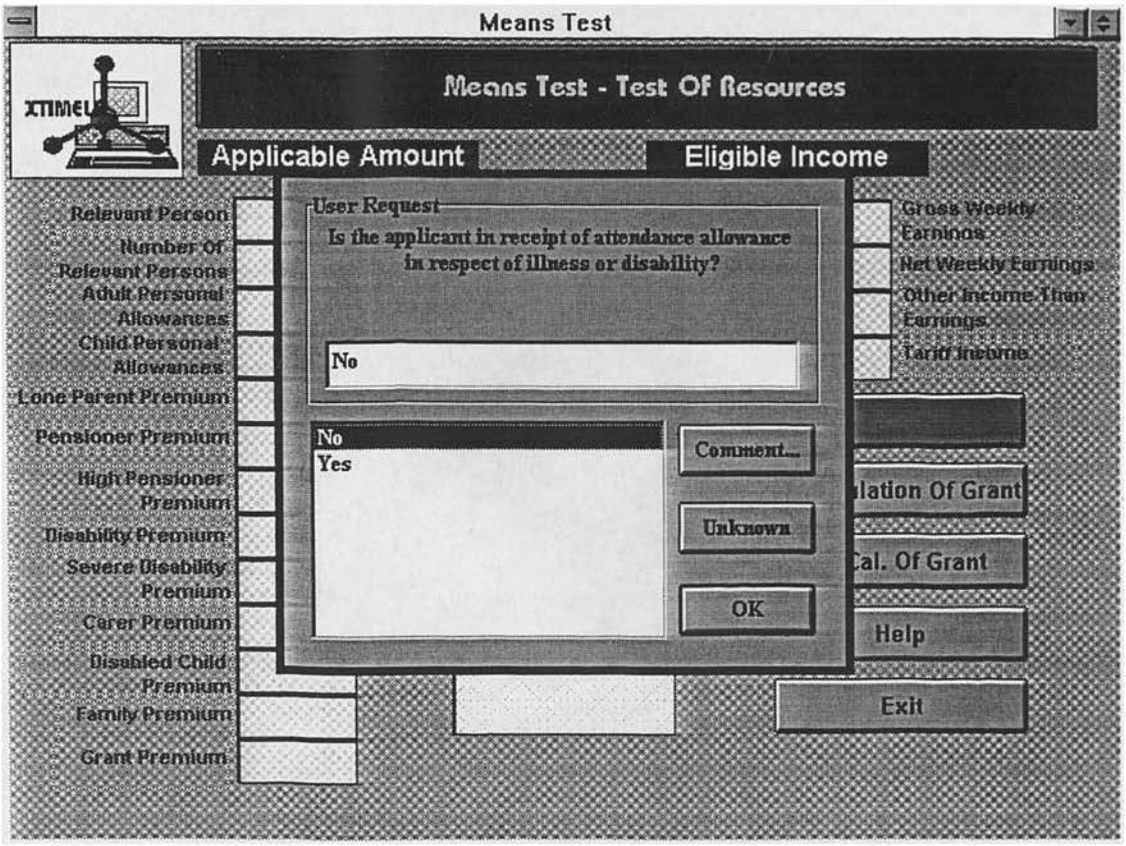
APPLICABLE AMOUNT



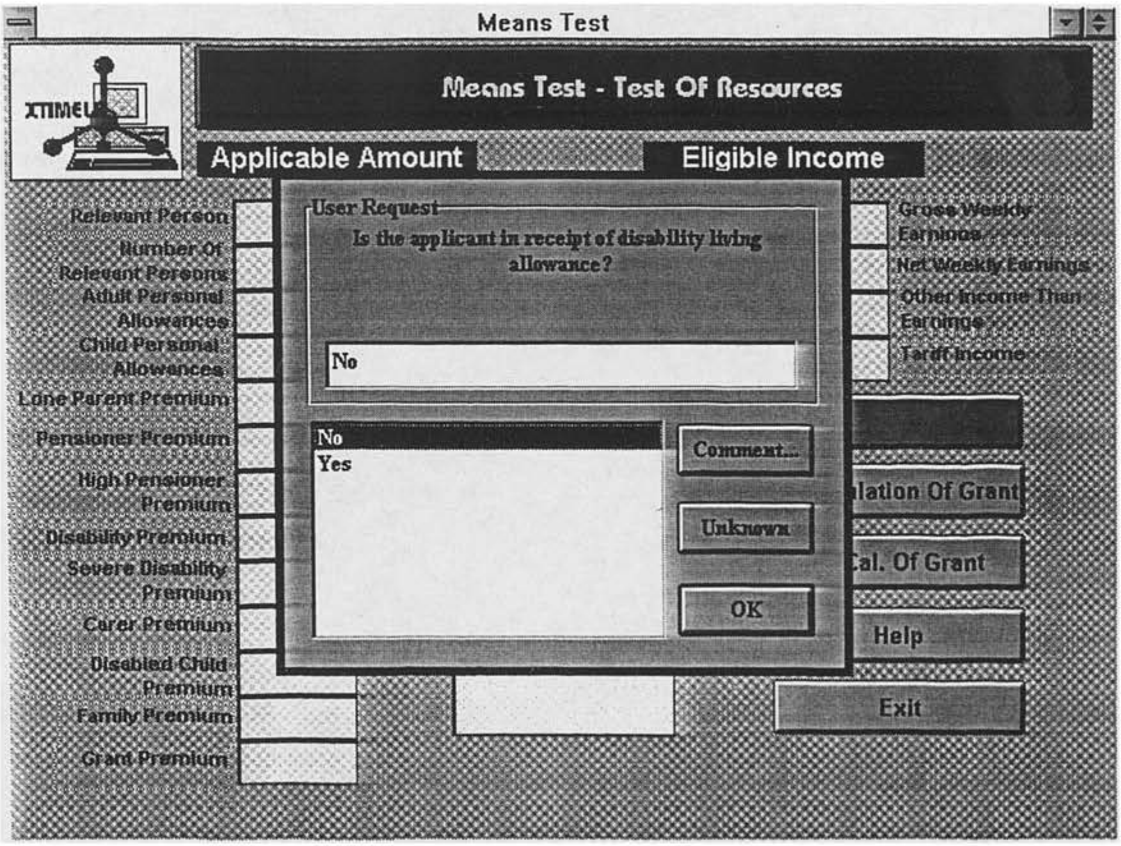
Screen 185: Input information for the test of resources



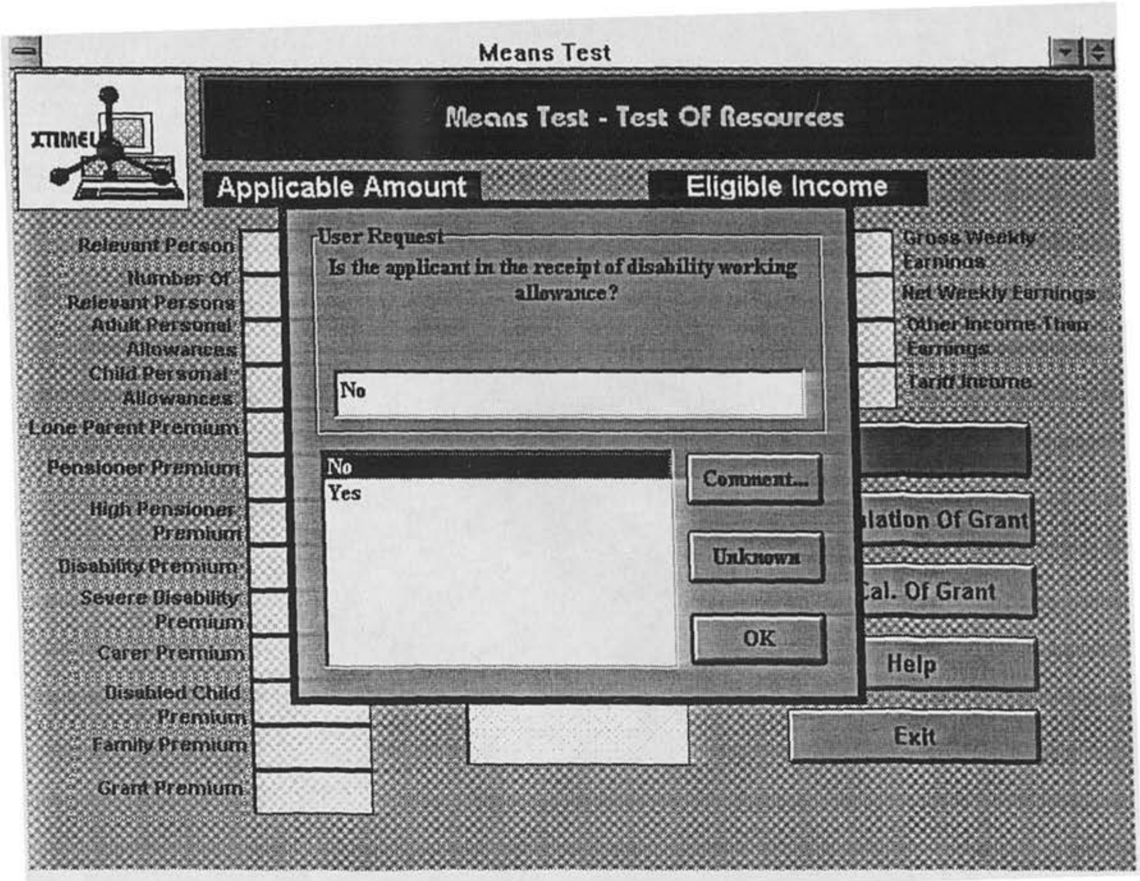
Screen 186: Input information for the test of resources



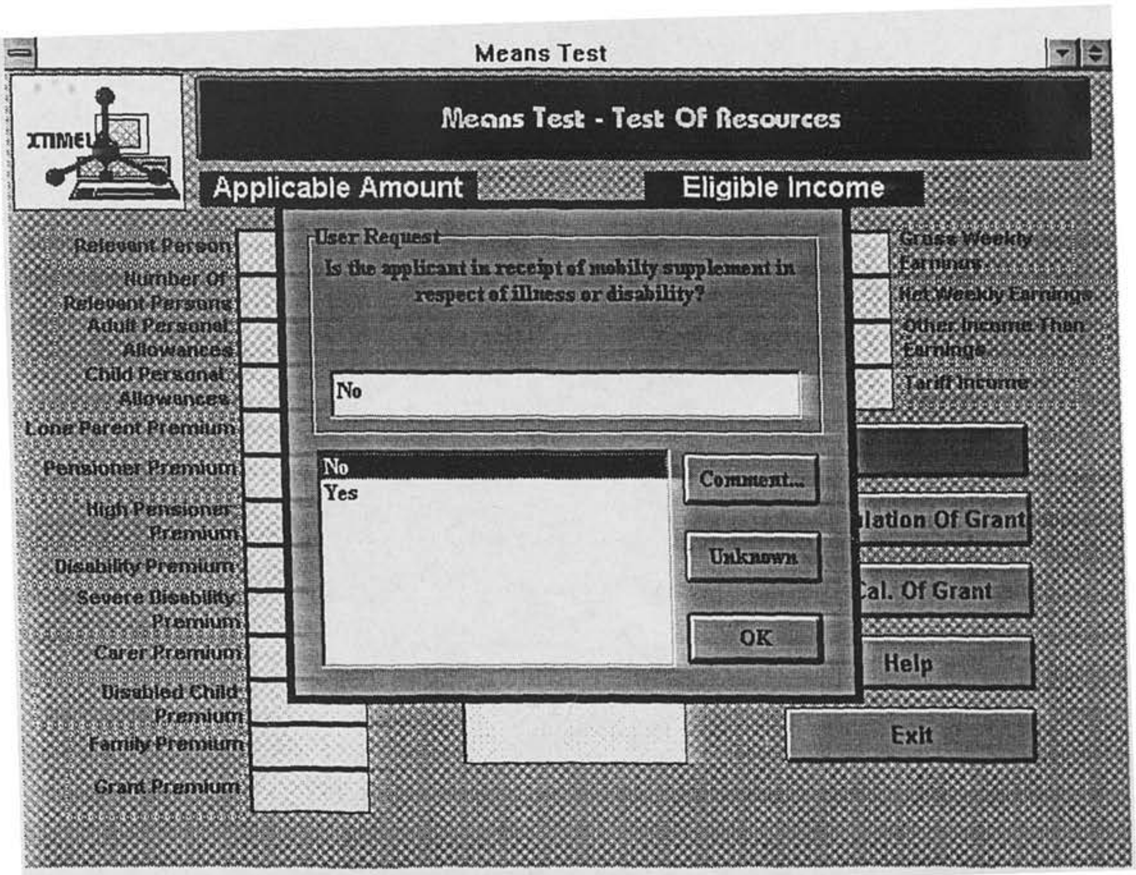
Screen 187: Input information for the test of resources



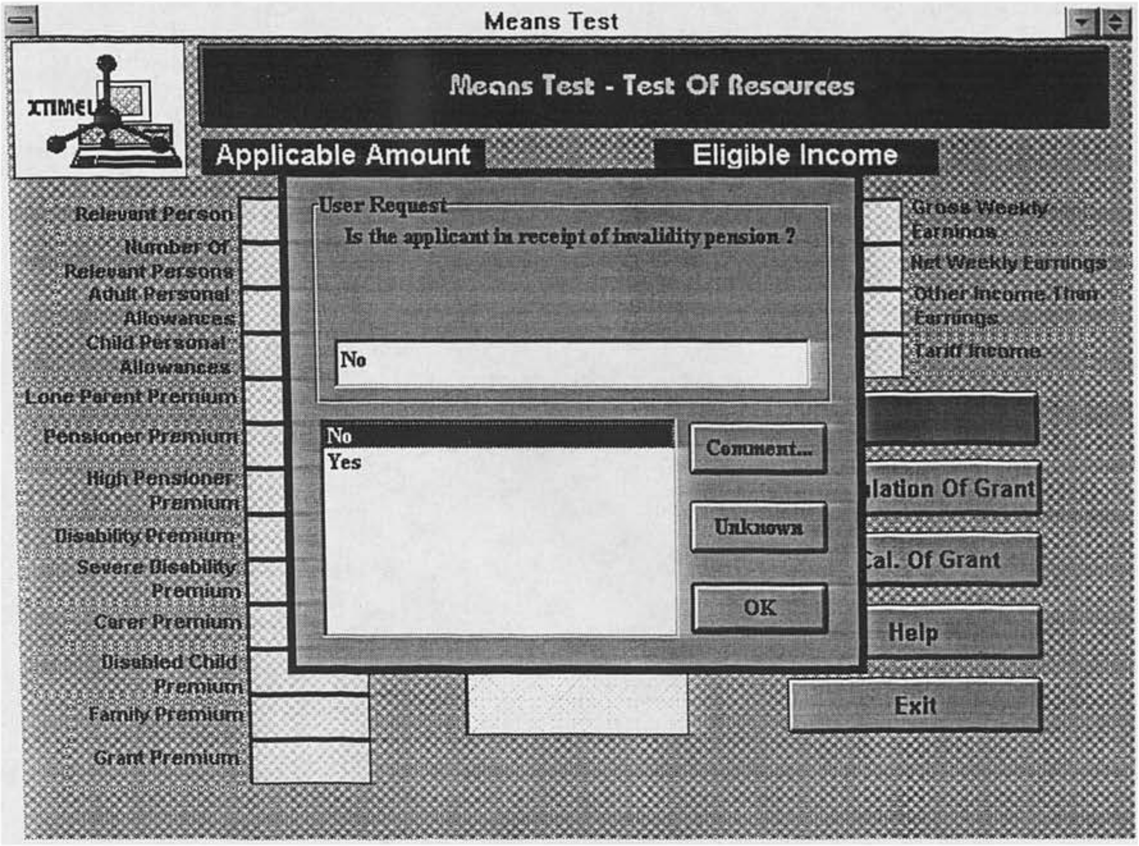
Screen 188: Input information for the test of resources



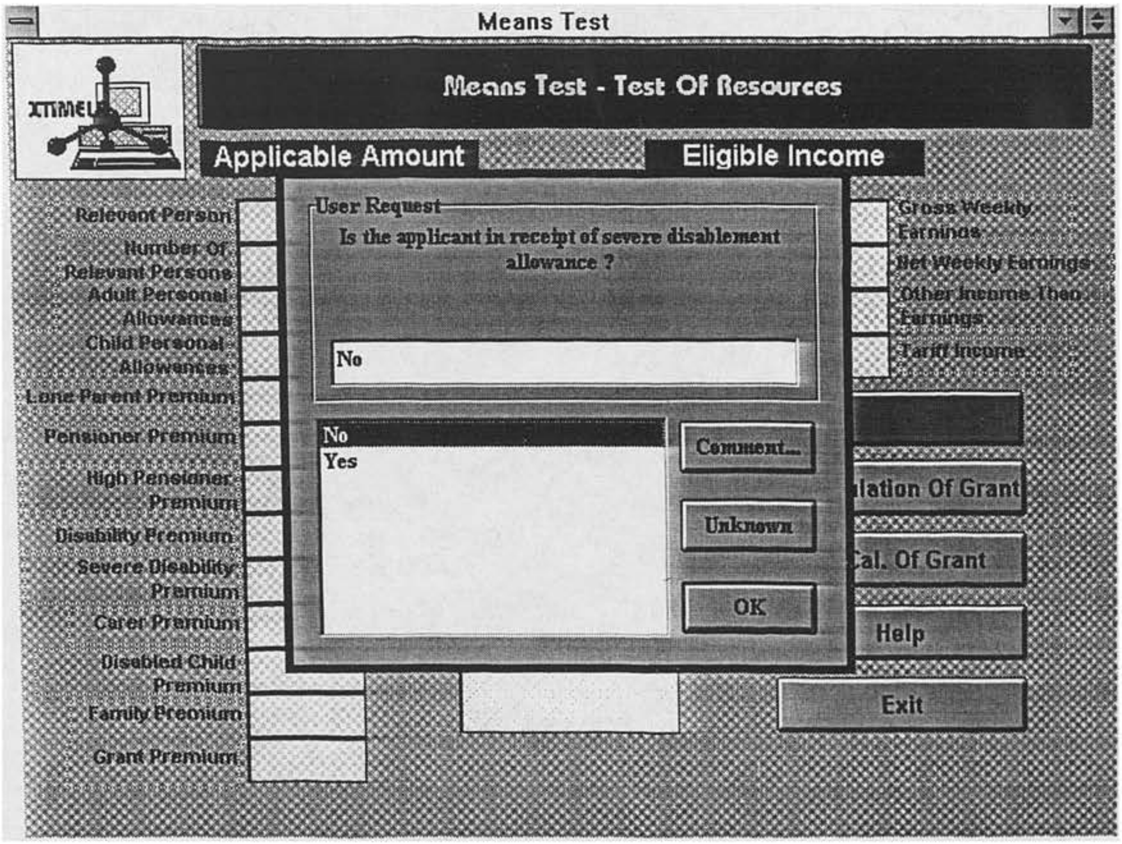
Screen 189: Input information for the test of resources



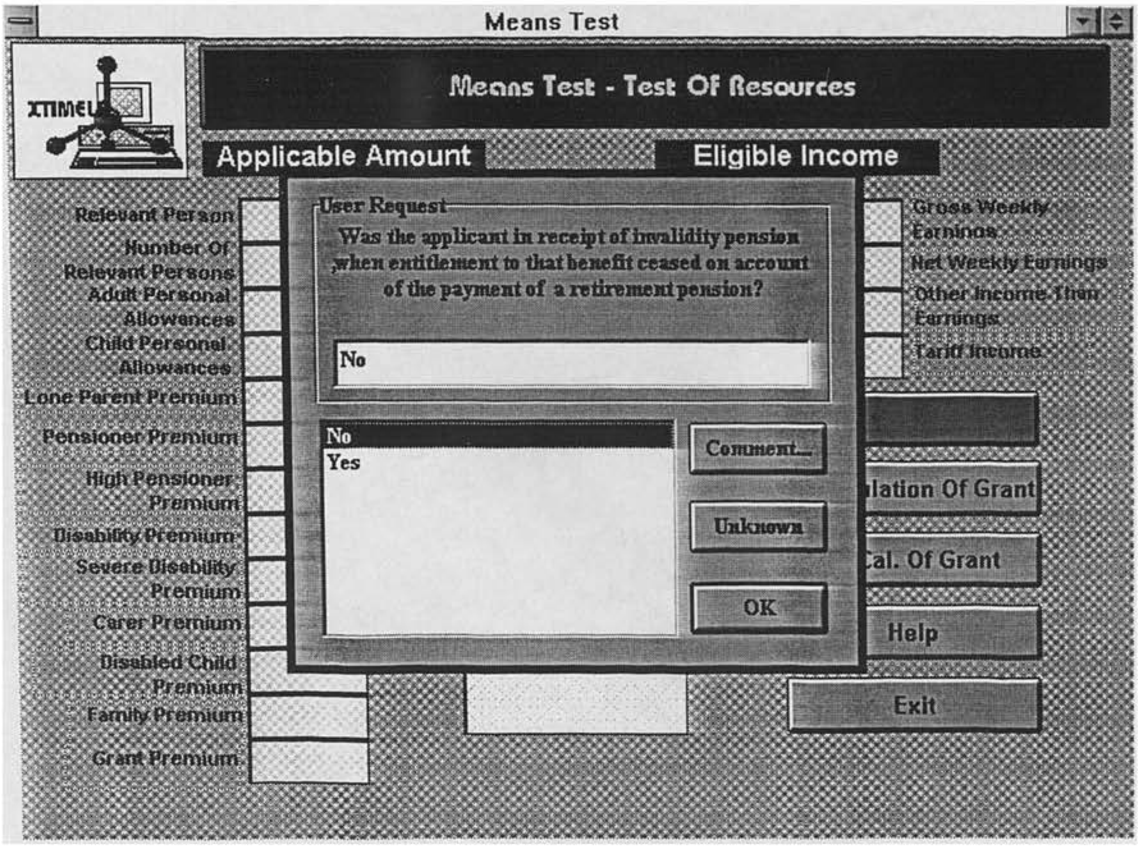
Screen 190: Input information for the test of resources



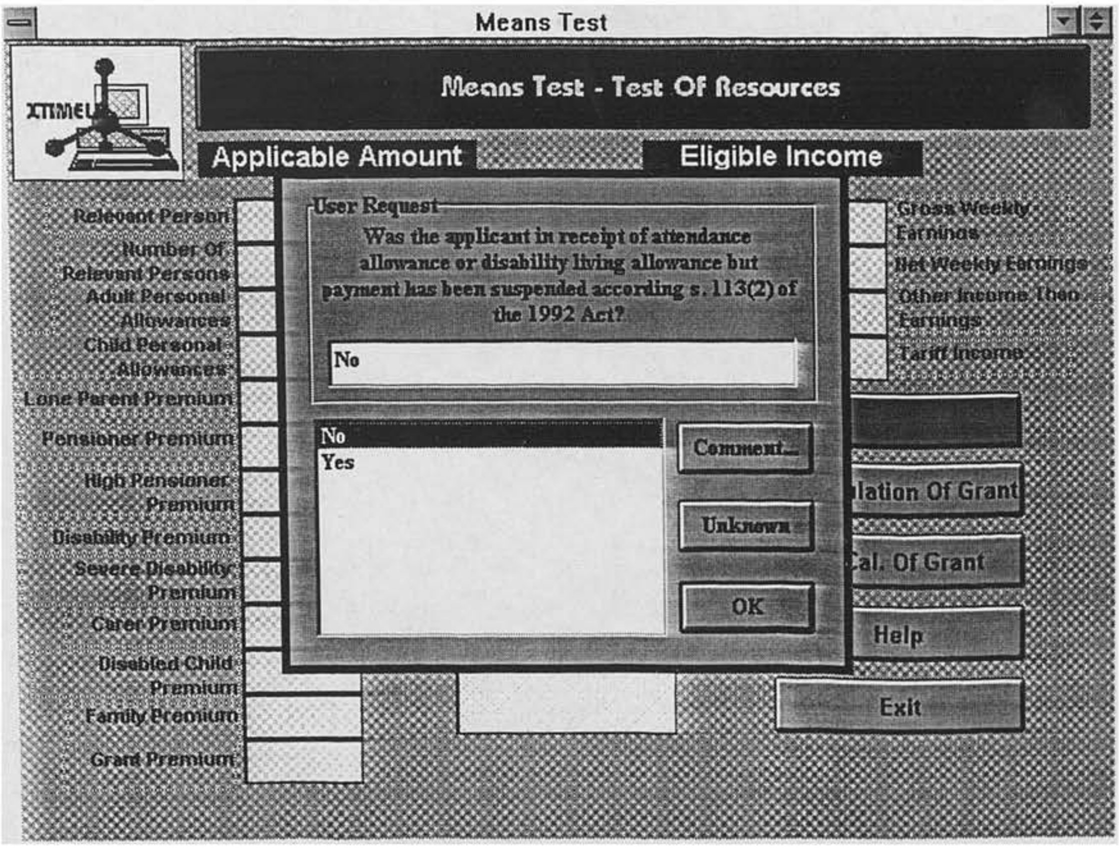
Screen 191: Input information for the test of resources



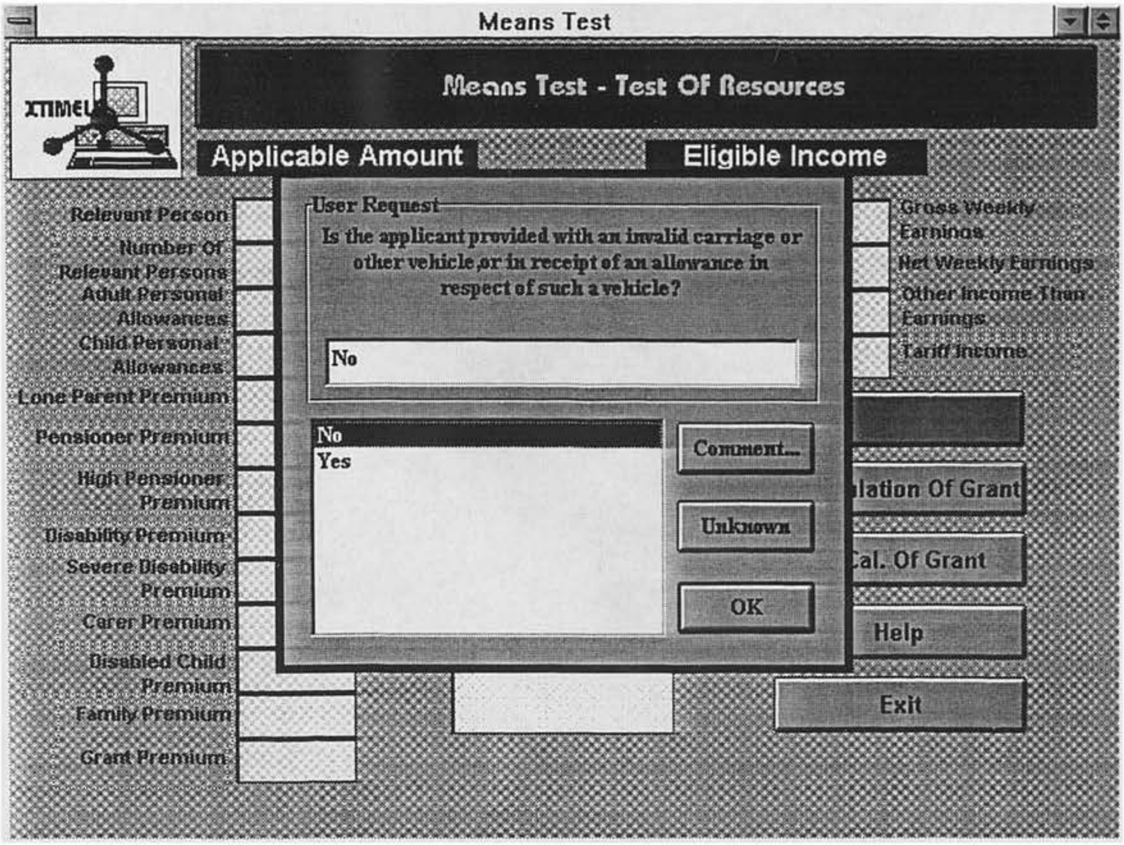
Screen 192: Input information for the test of resources



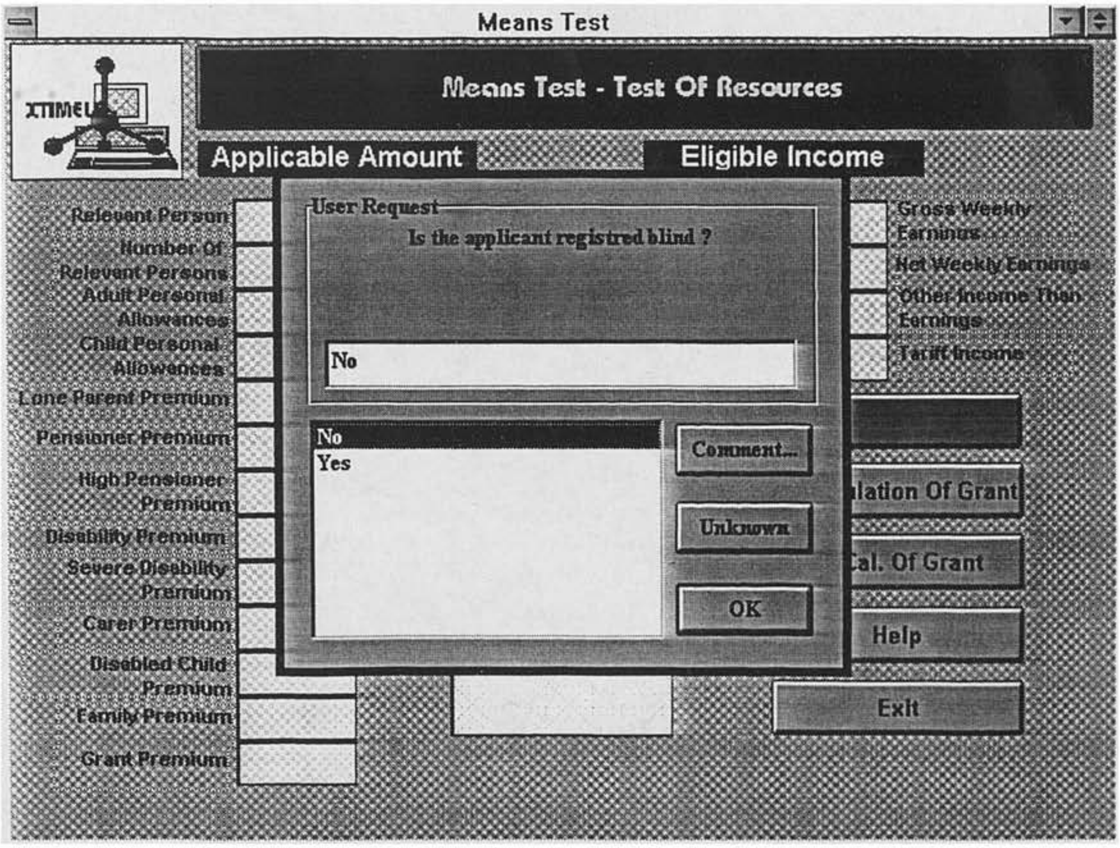
Screen 193: Input information for the test of resources



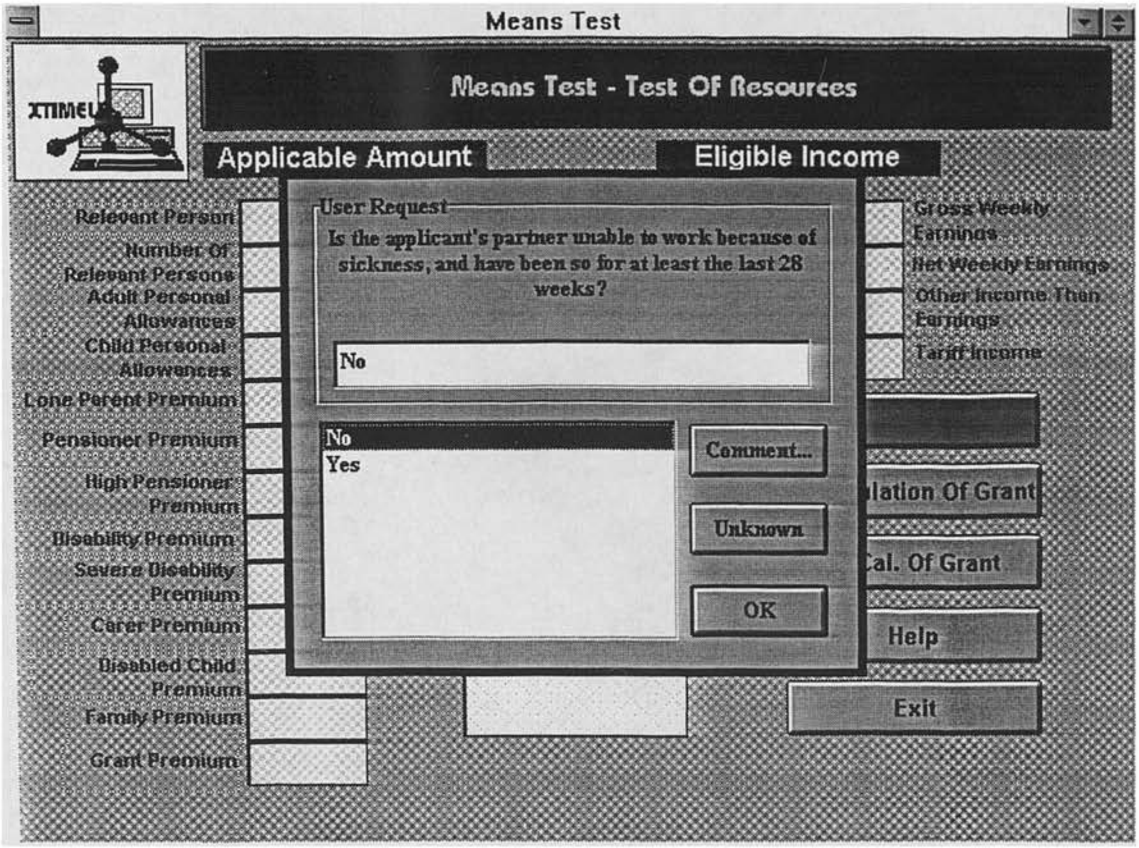
Screen 194: Input information for the test of resources



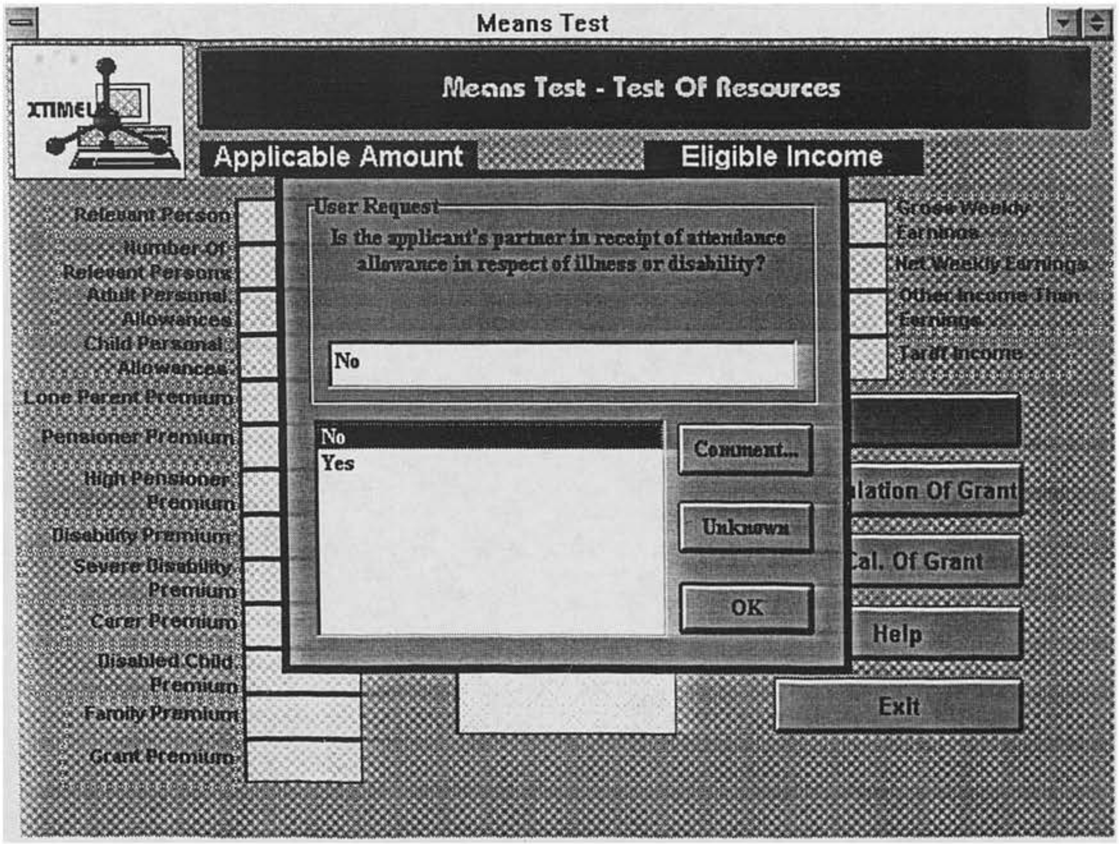
Screen 195: Input information for the test of resources



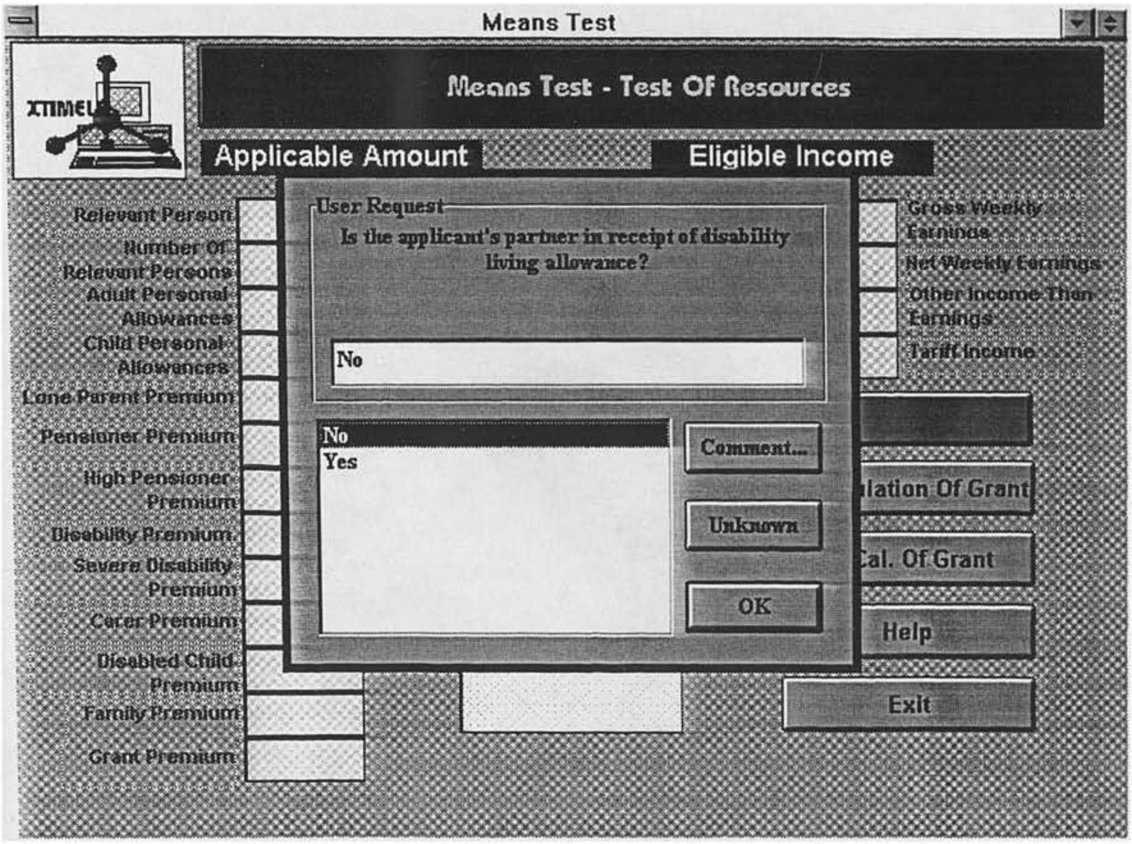
Screen 196: Input information for the test of resources



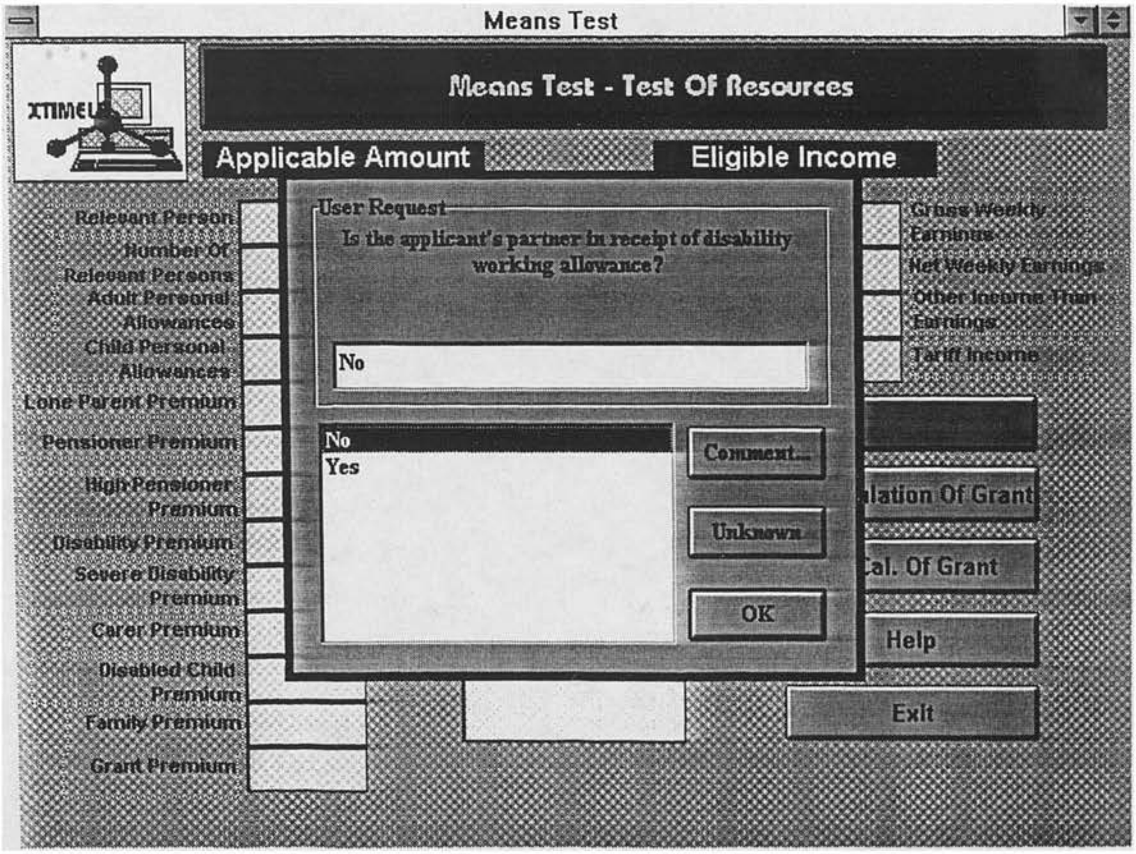
Screen 197: Input information for the test of resources



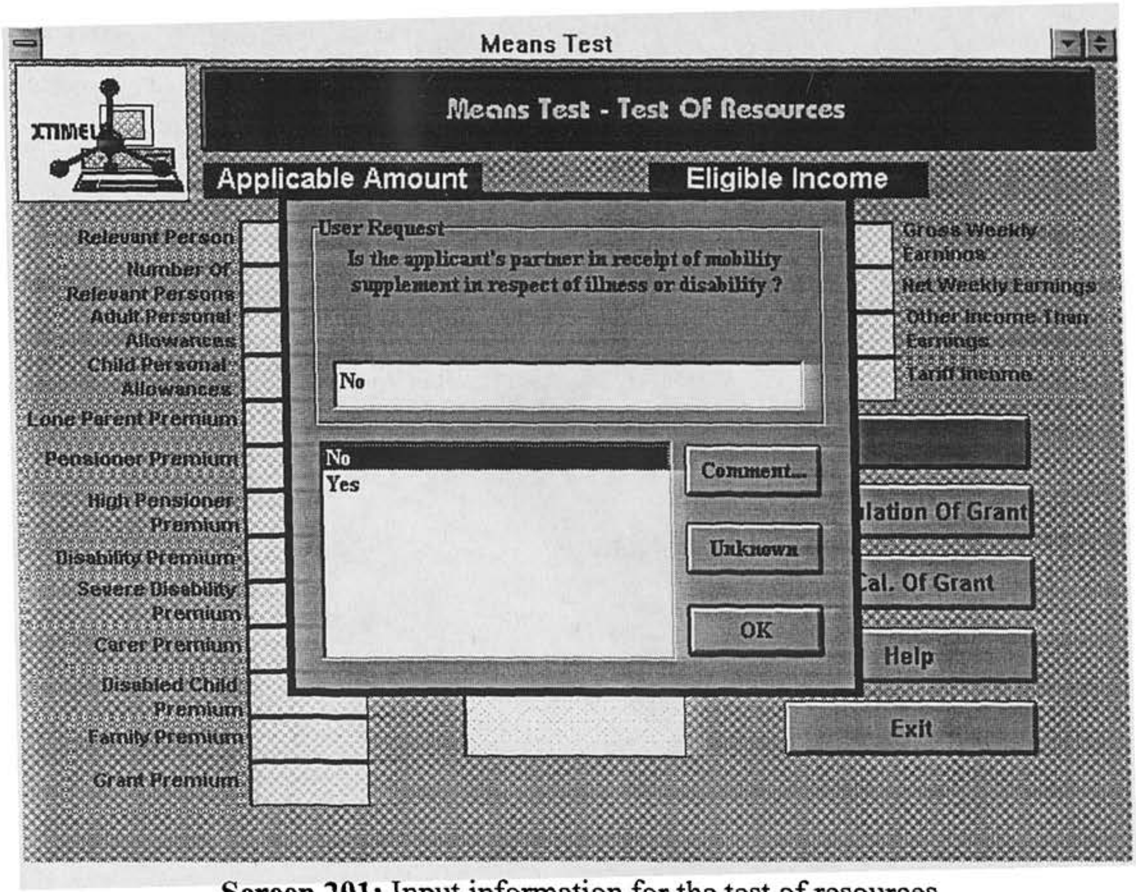
Screen 198: Input information for the test of resources



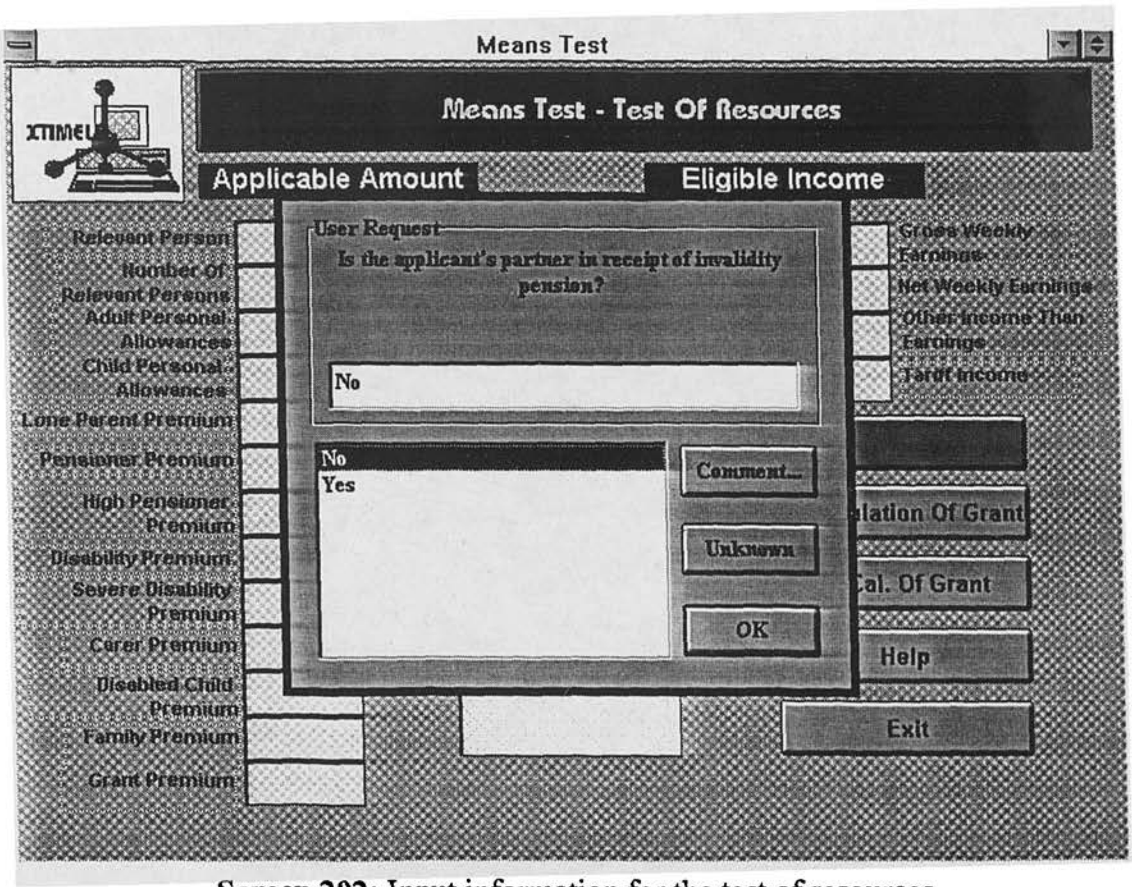
Screen 199: Input information for the test of resources



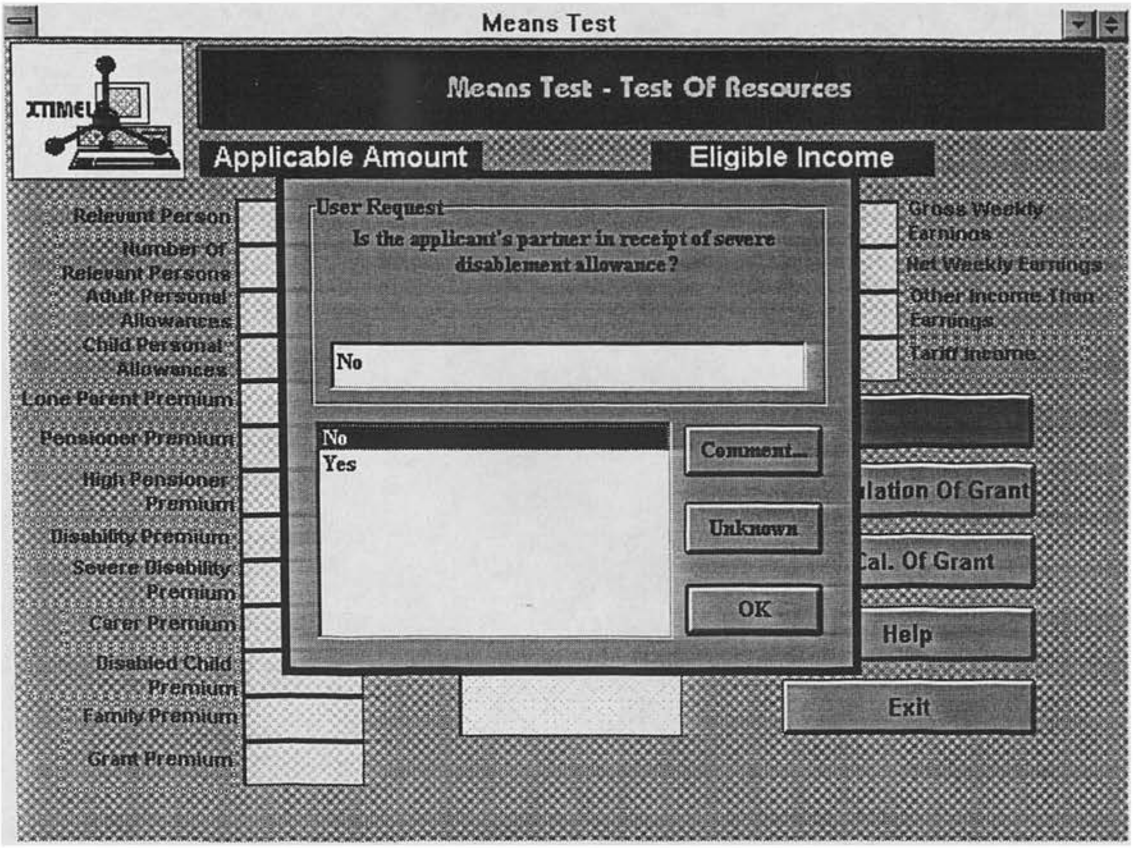
Screen 200: Input information for the test of resources



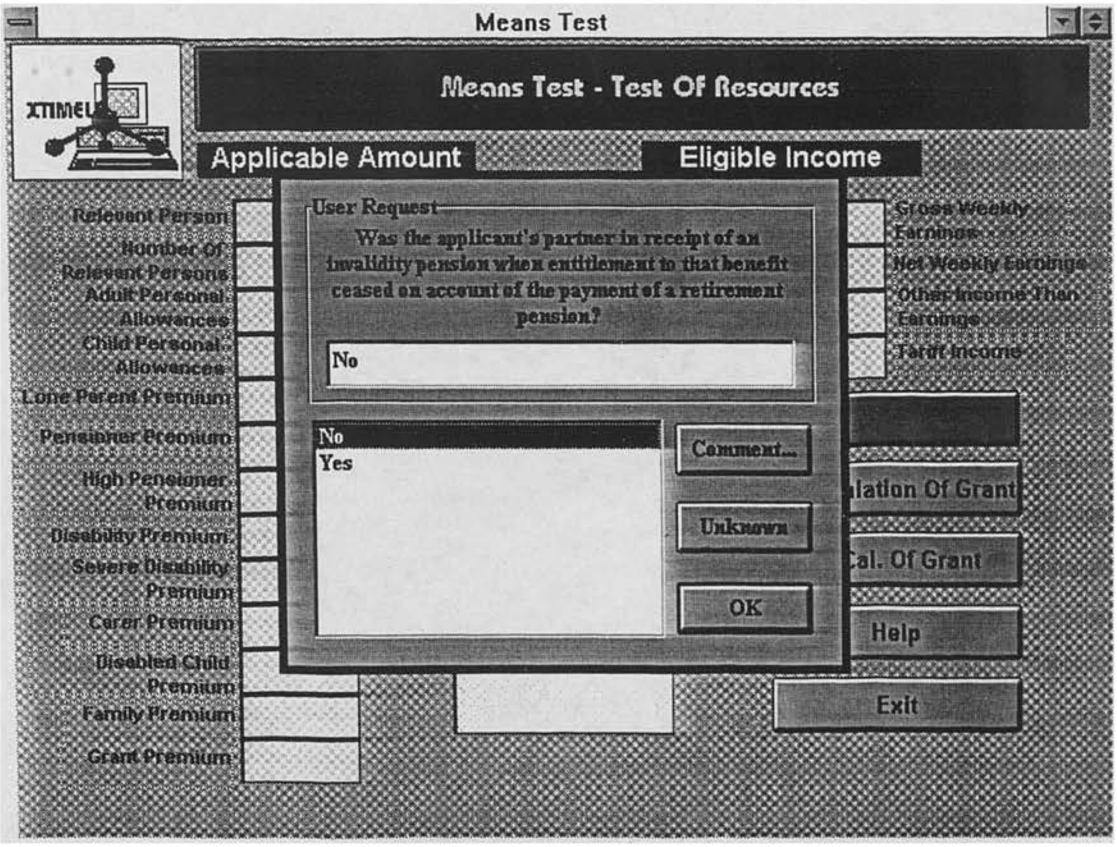
Screen 201: Input information for the test of resources



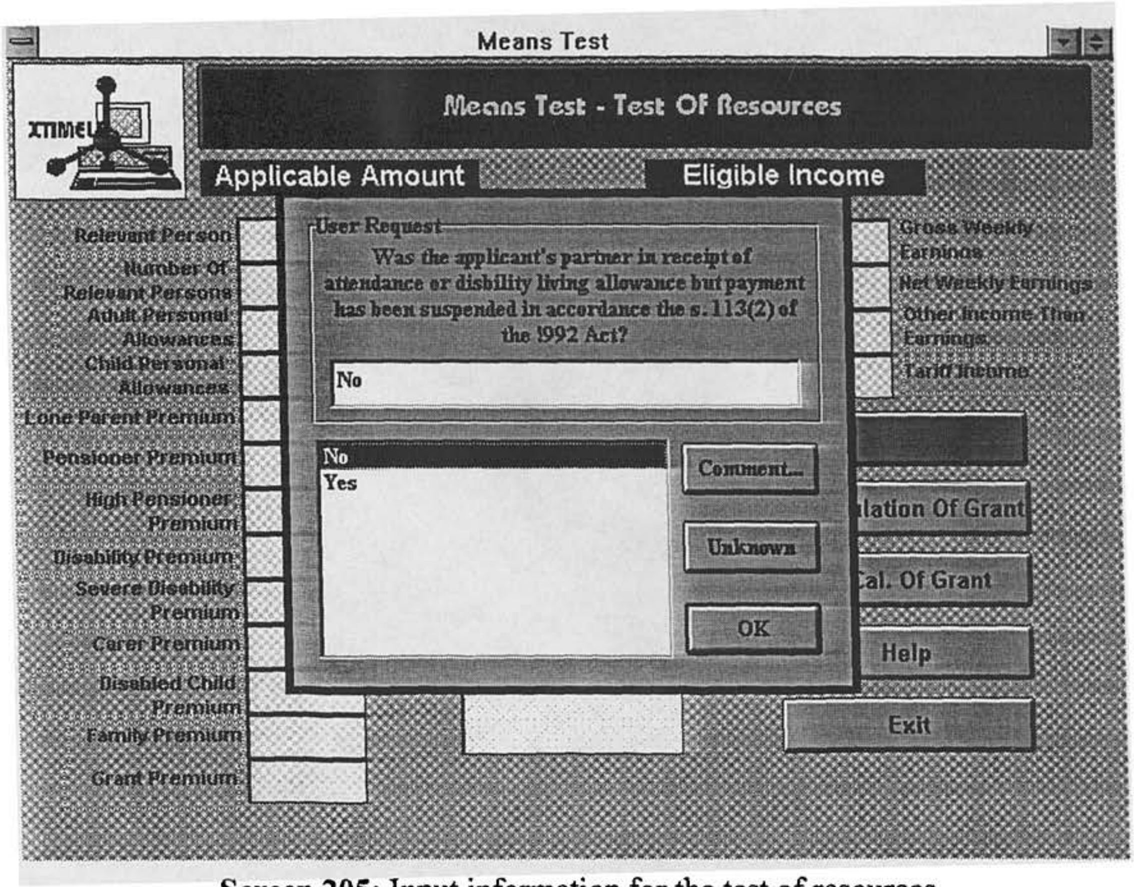
Screen 202: Input information for the test of resources



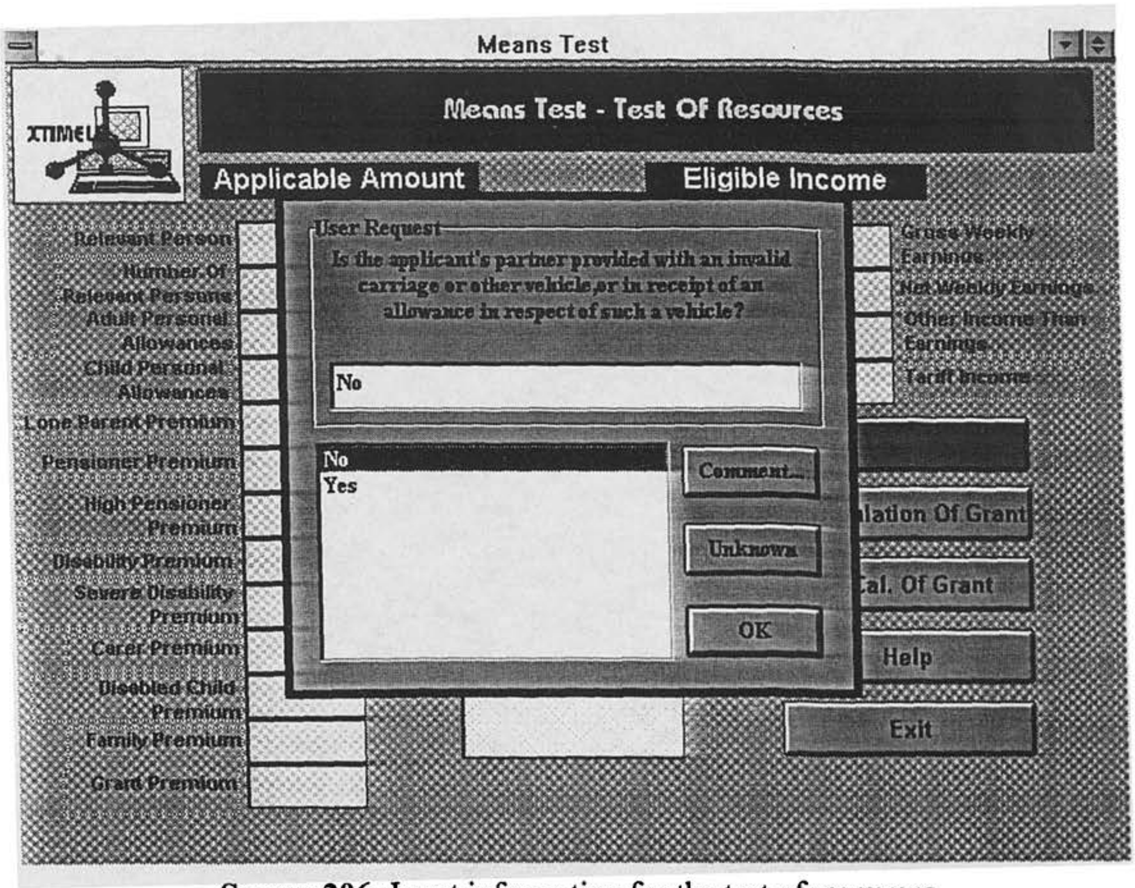
Screen 203: Input information for the test of resources



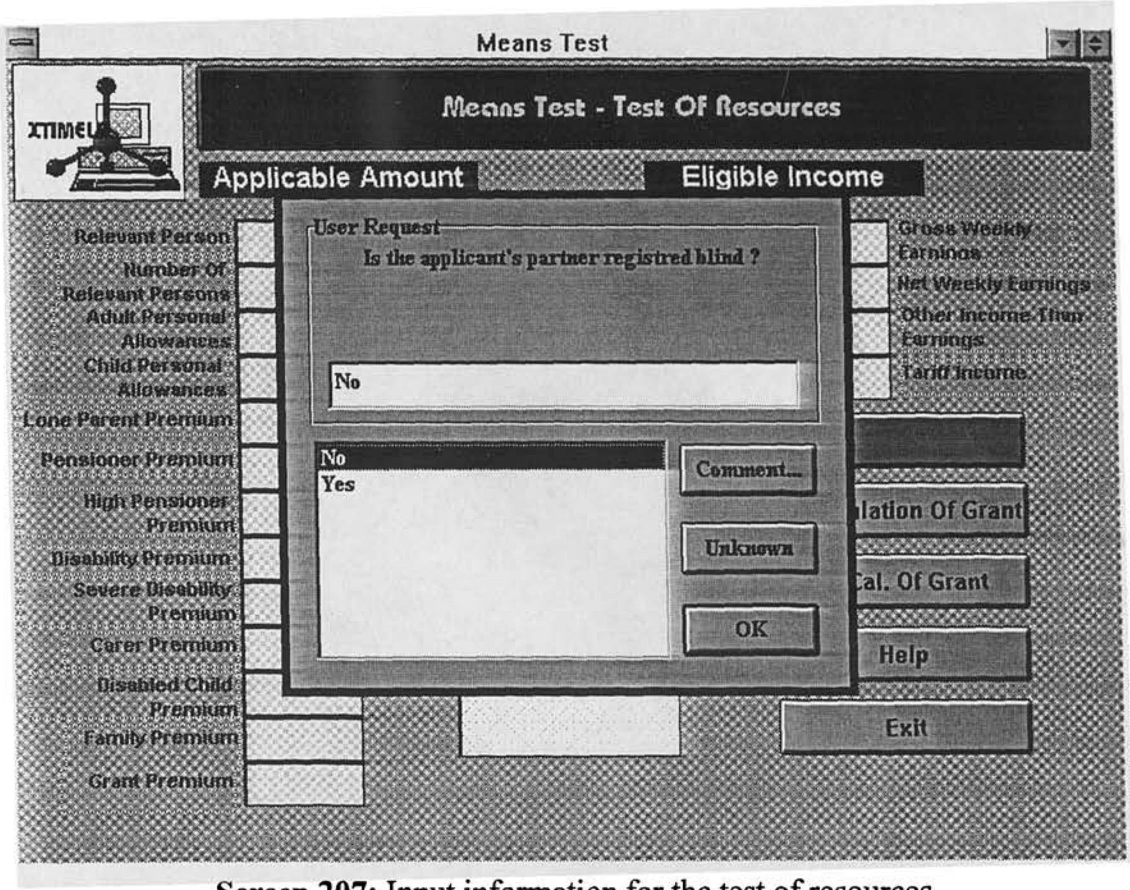
Screen 204: Input information for the test of resources



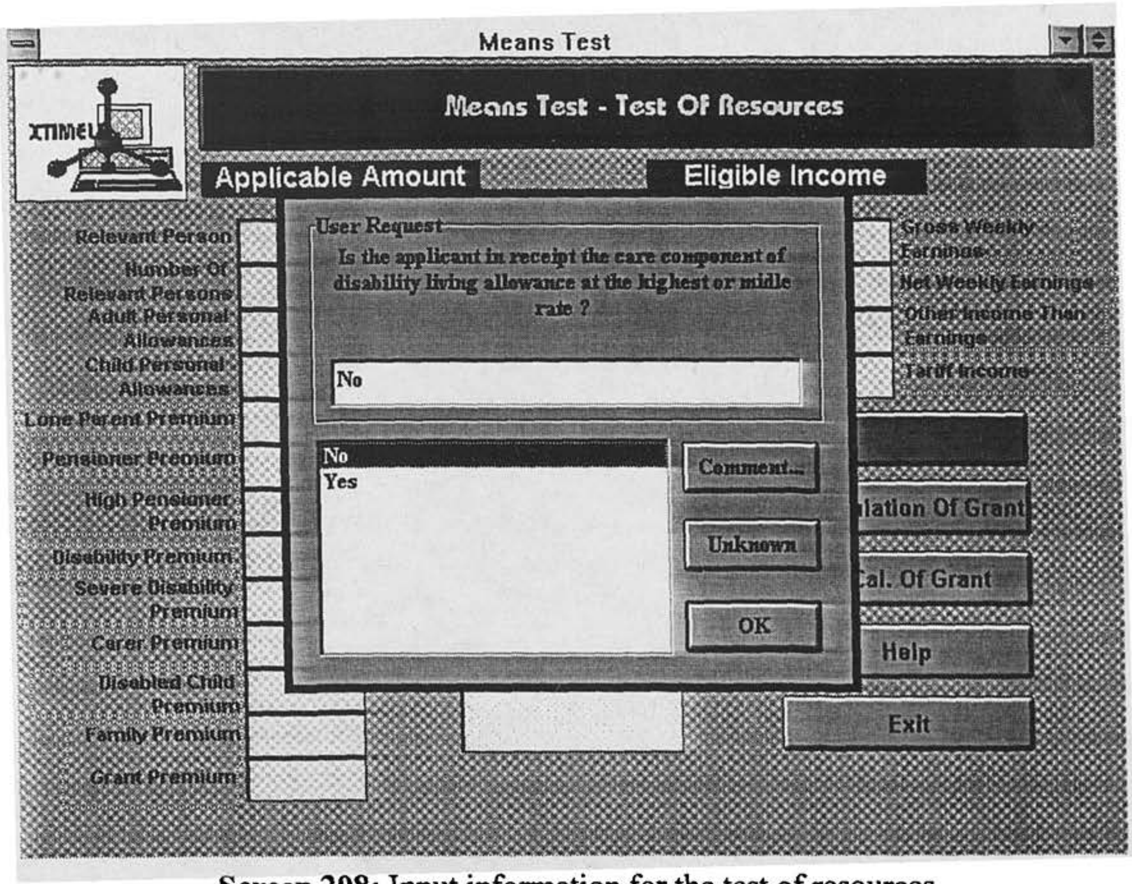
Screen 205: Input information for the test of resources



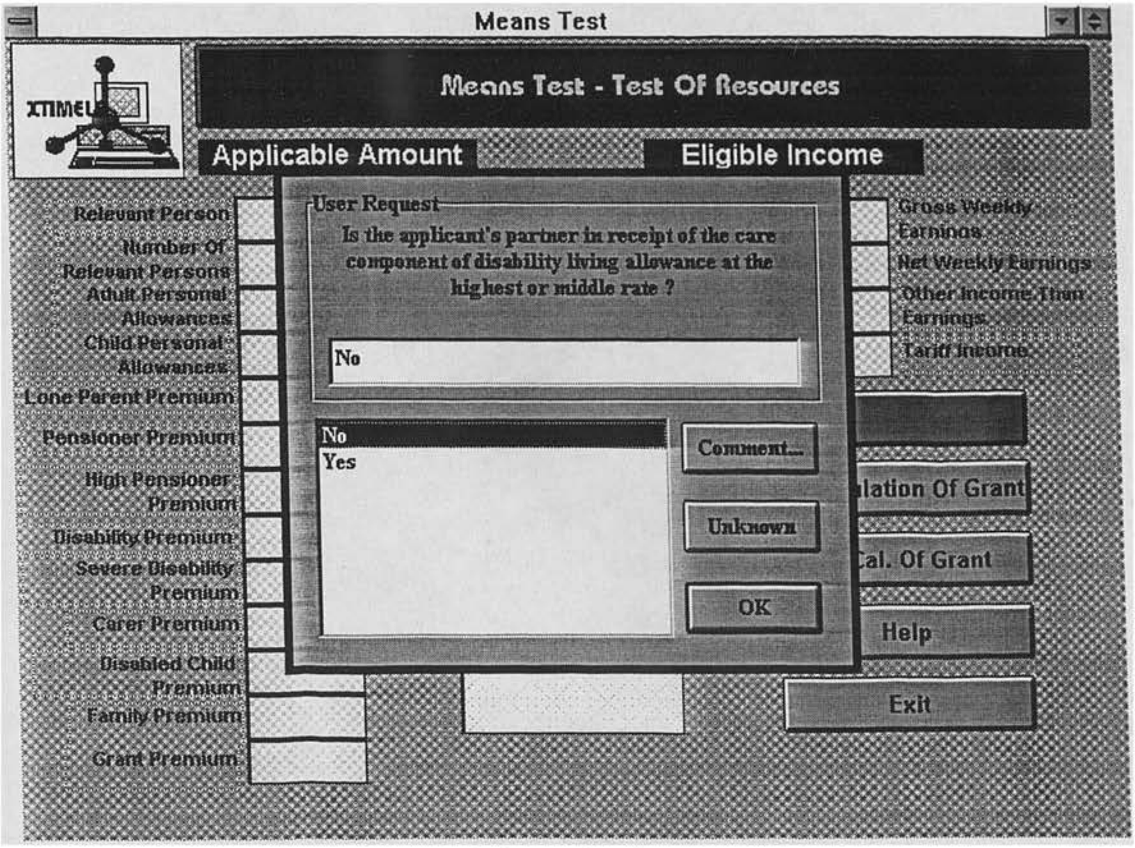
Screen 206: Input information for the test of resources



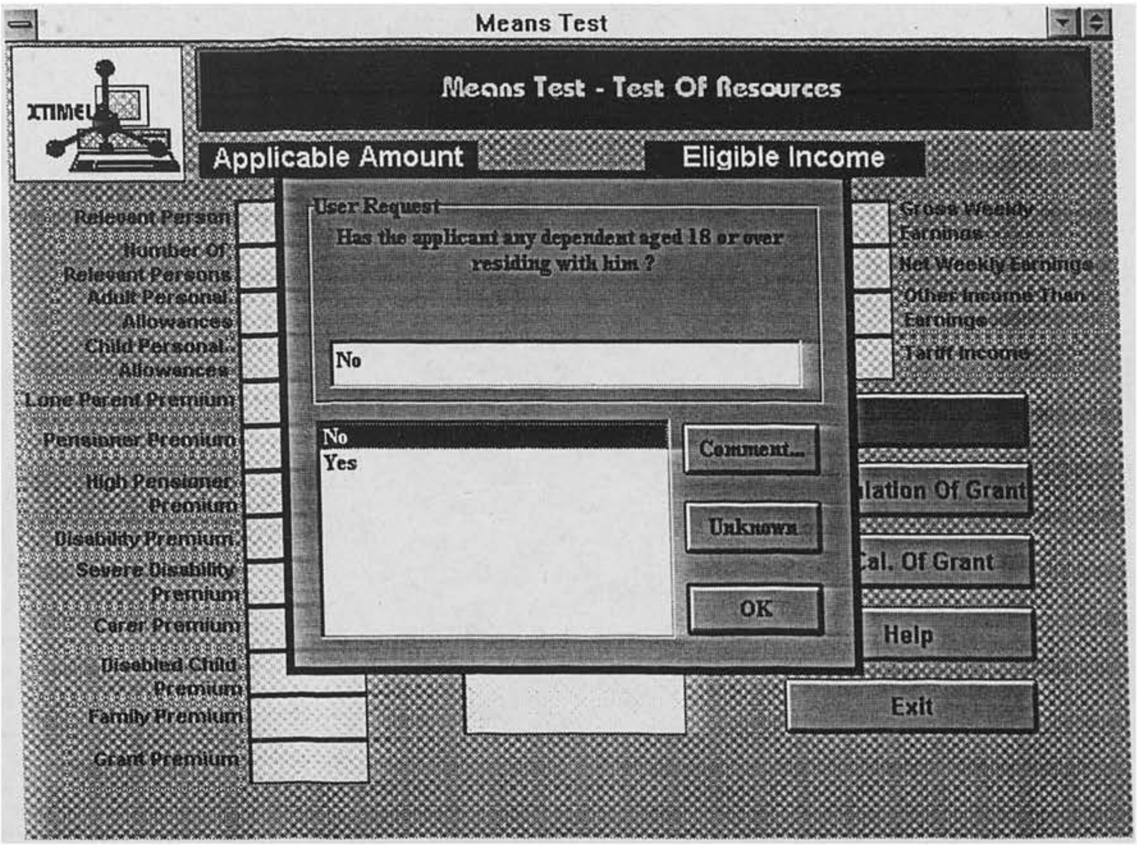
Screen 207: Input information for the test of resources



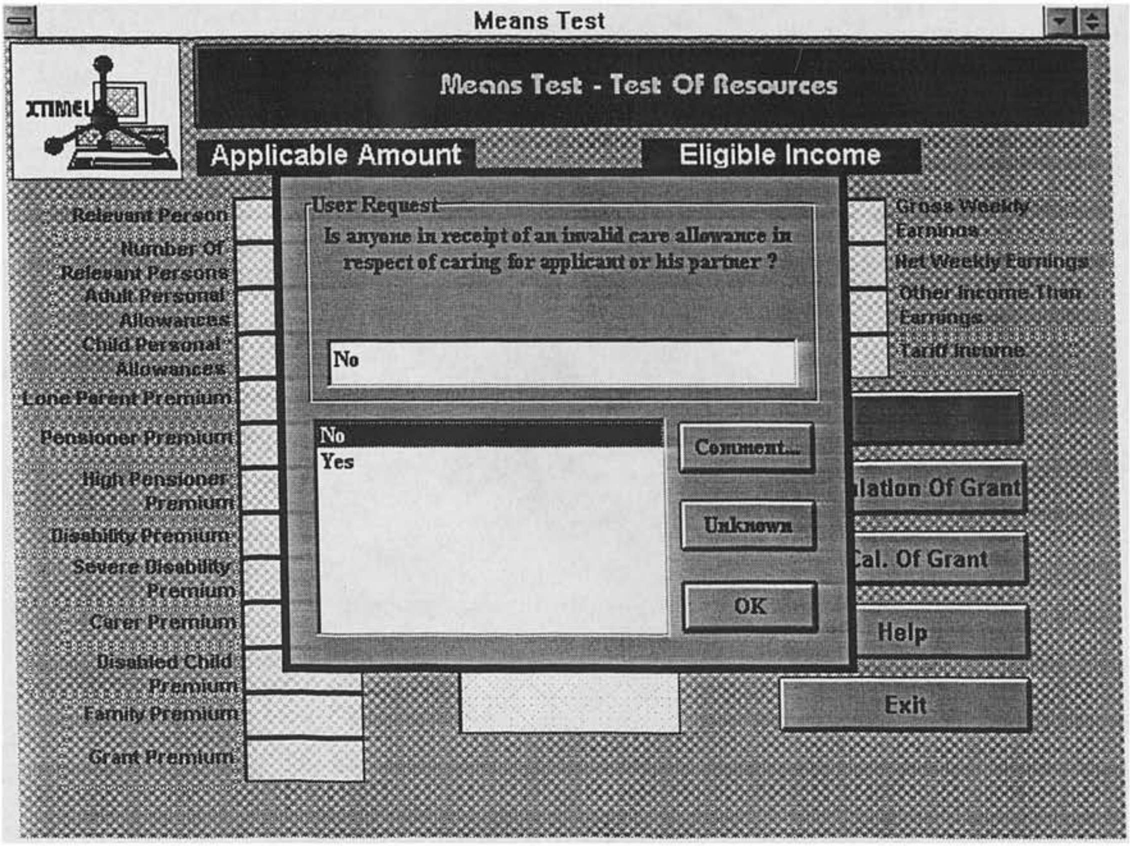
Screen 208: Input information for the test of resources



Screen 209: Input information for the test of resources

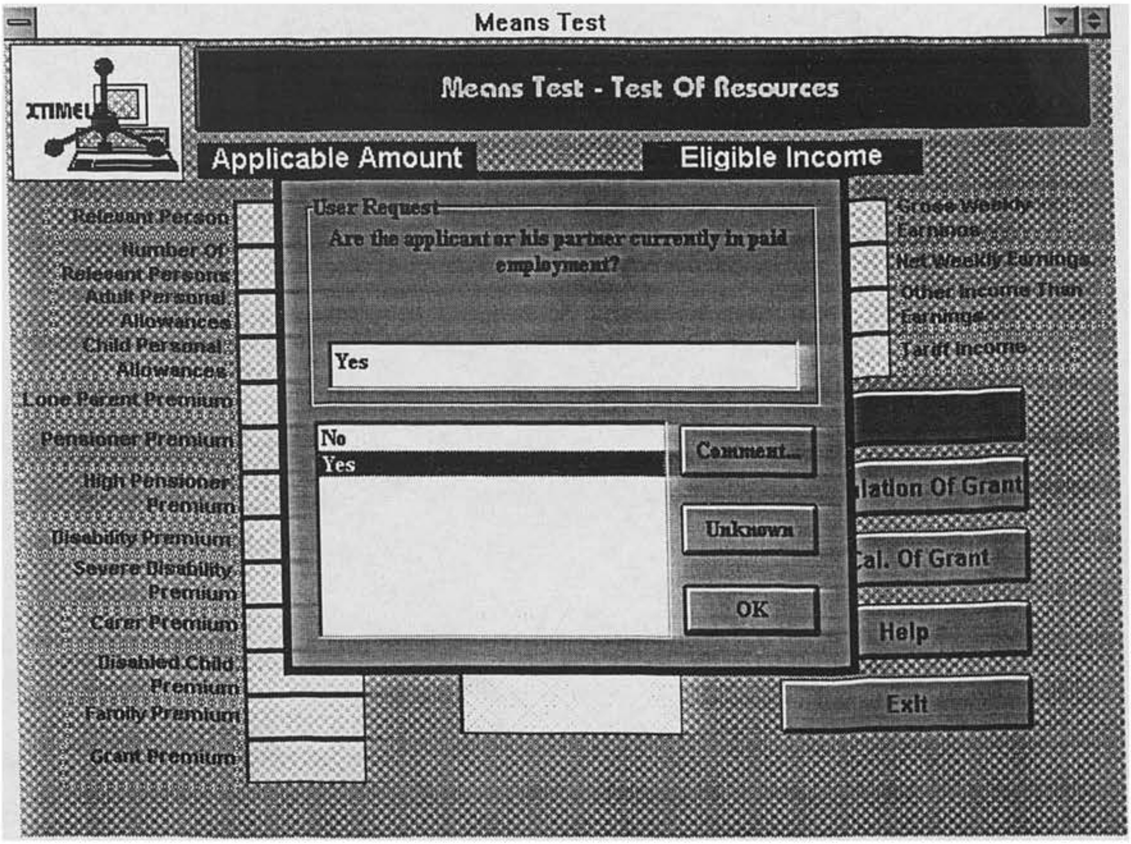


Screen 210: Input information for the test of resources

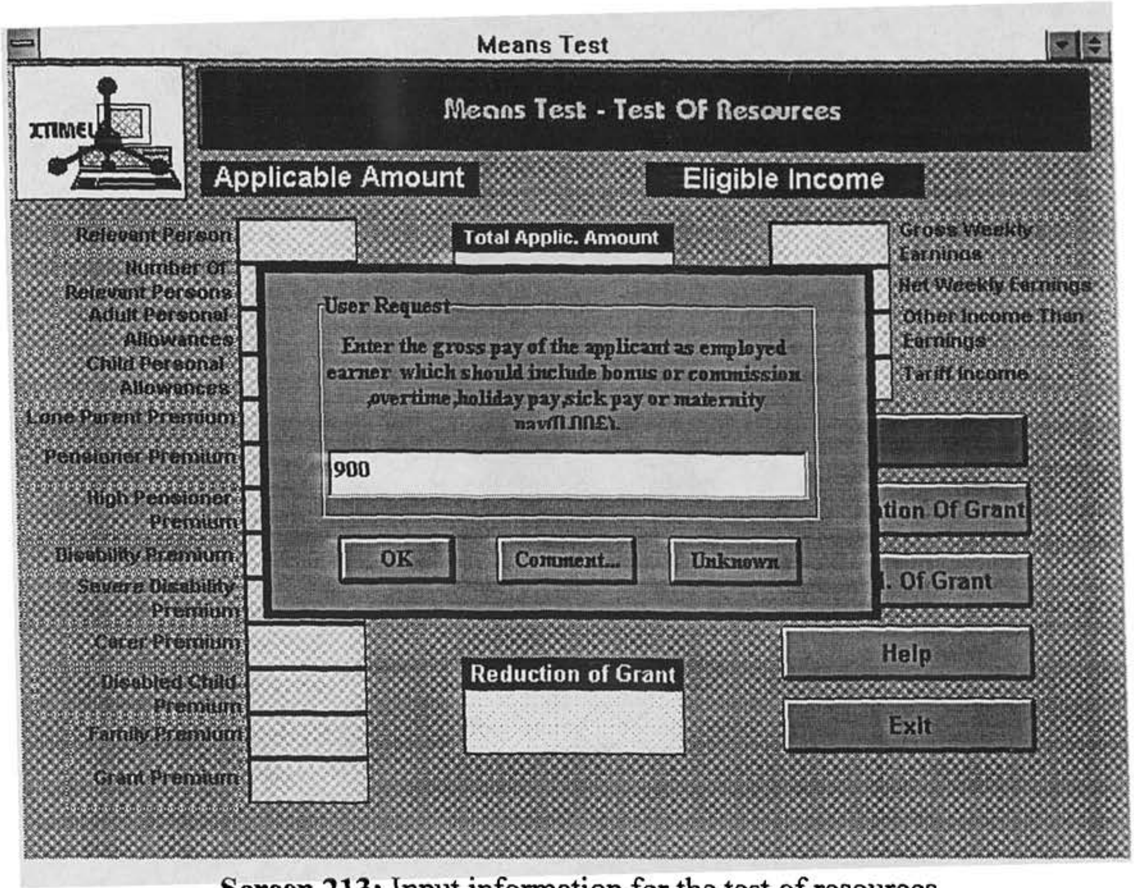


Screen 211: Input information for the test of resources

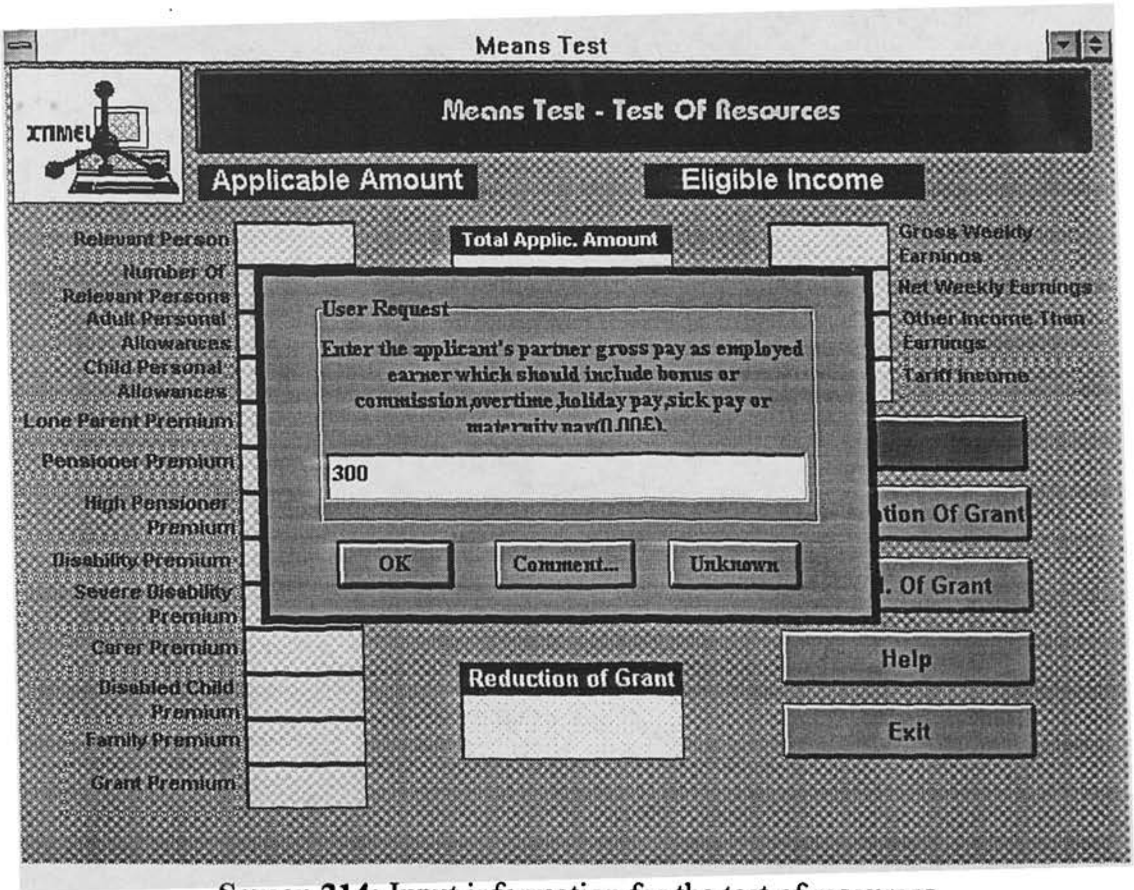
ELIGIBLE INCOME AND CAPITAL



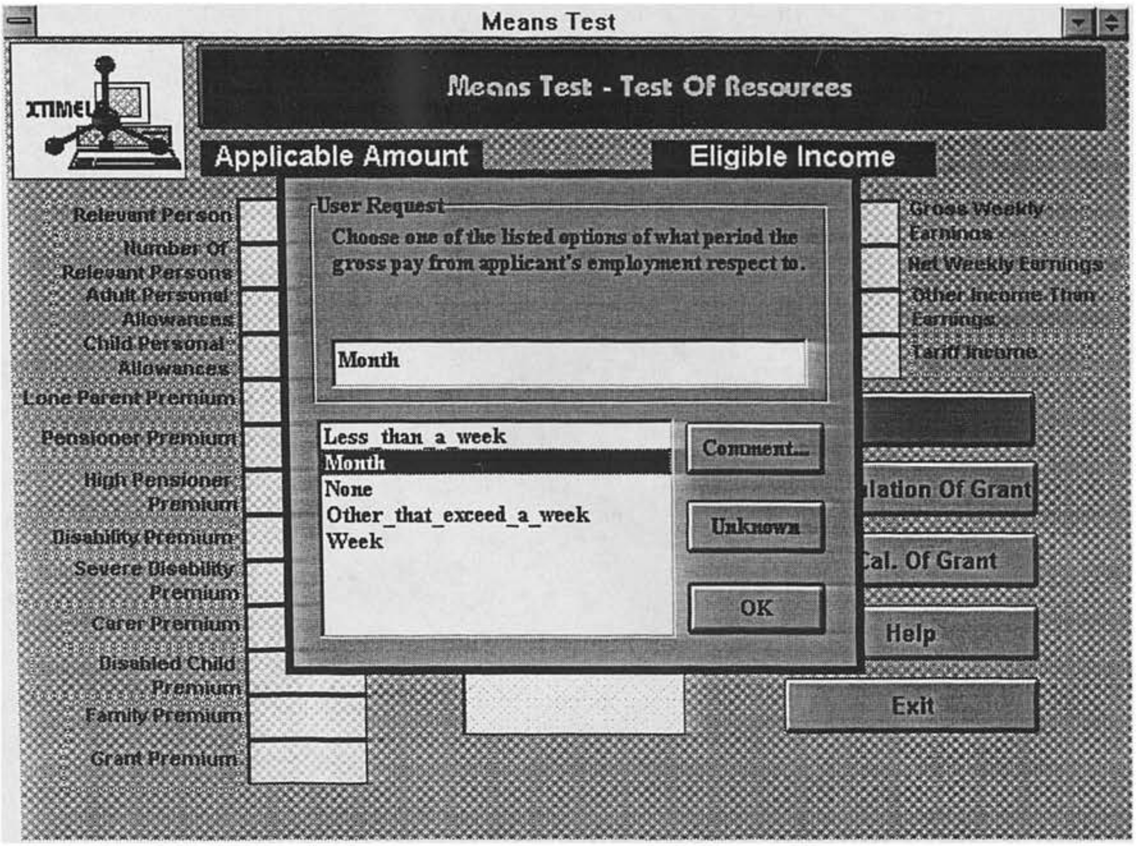
Screen 212: Input information for the test of resources



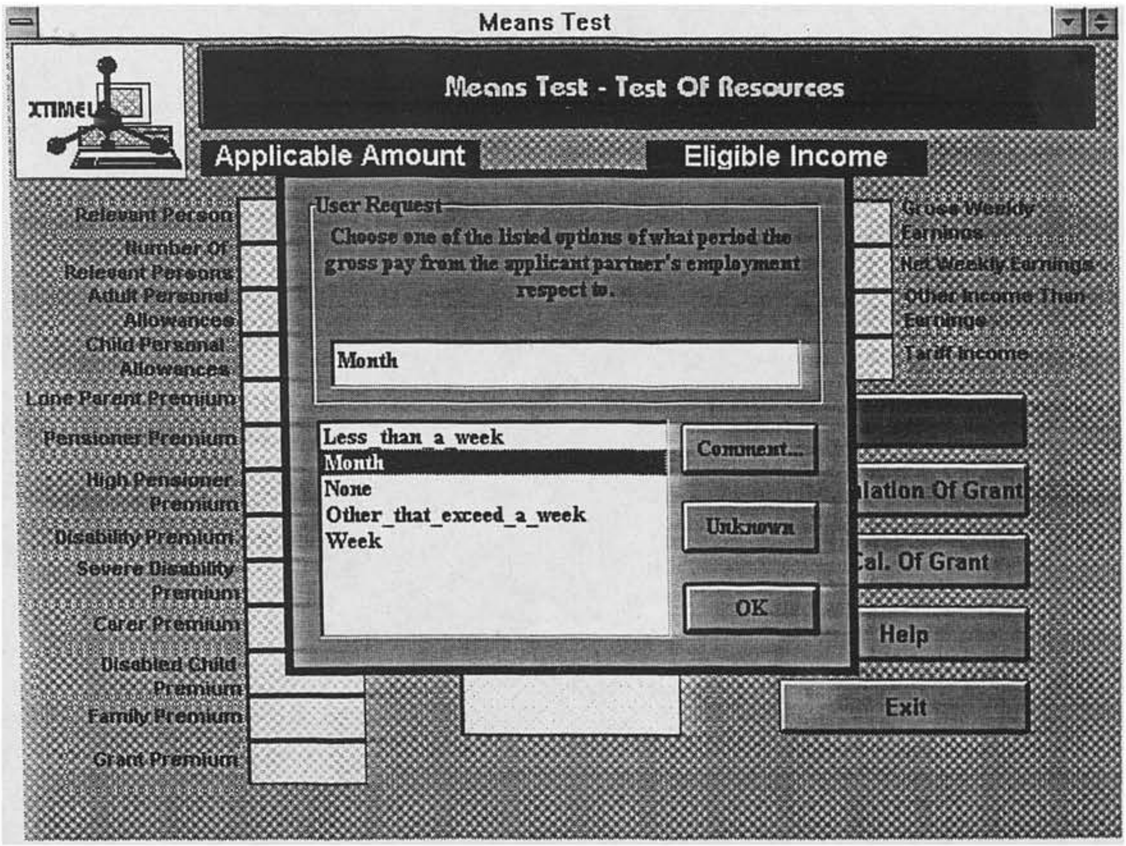
Screen 213: Input information for the test of resources



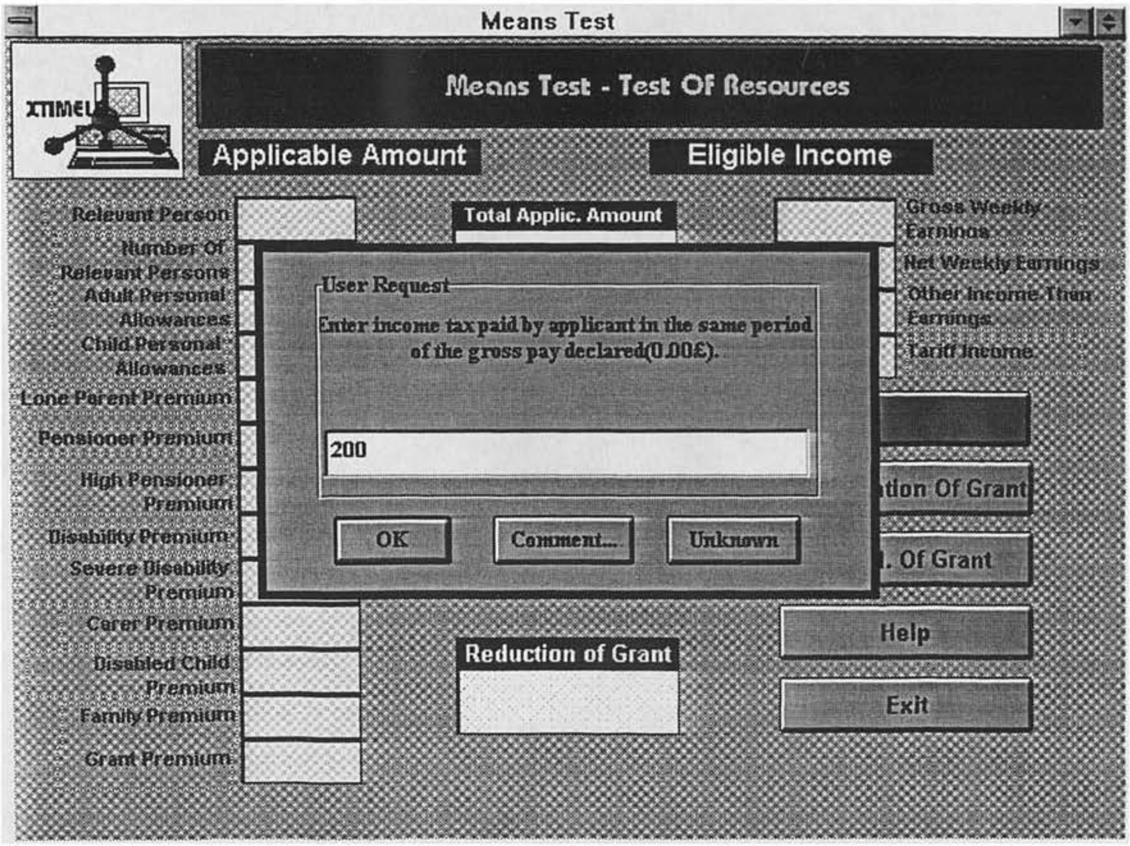
Screen 214: Input information for the test of resources



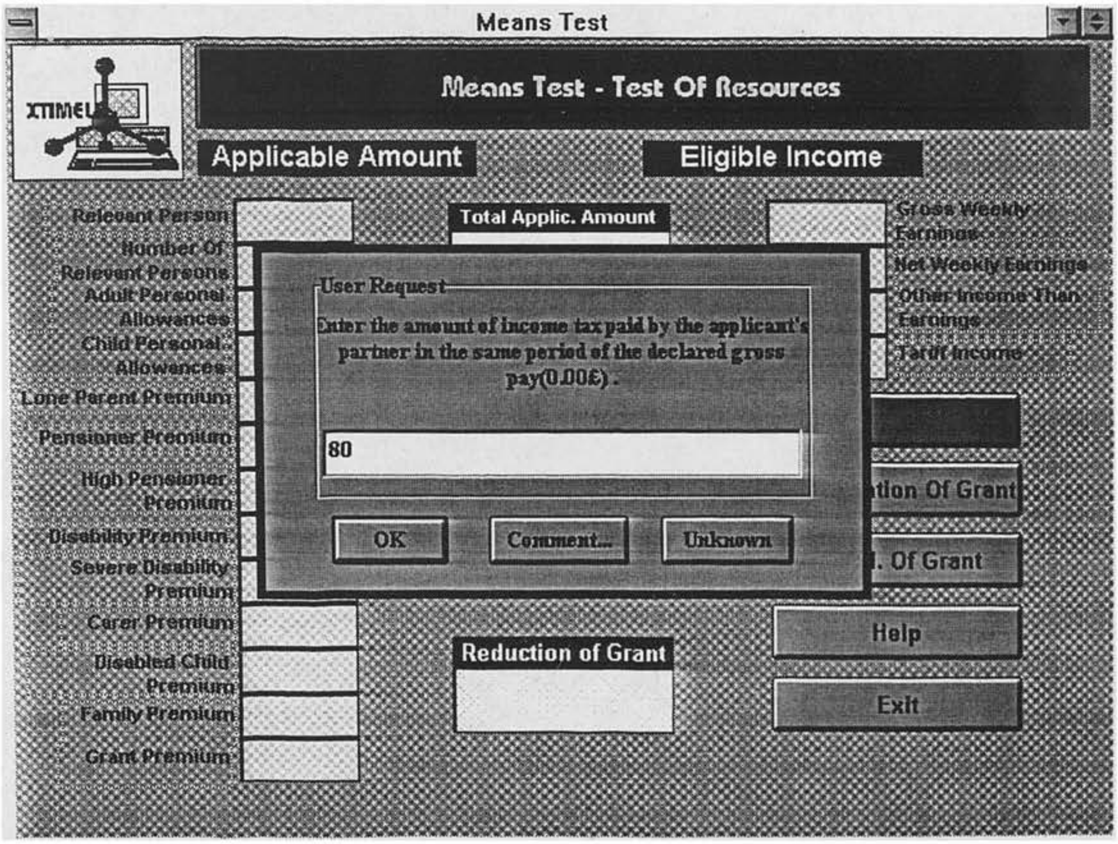
Screen 215: Input information for the test of resources



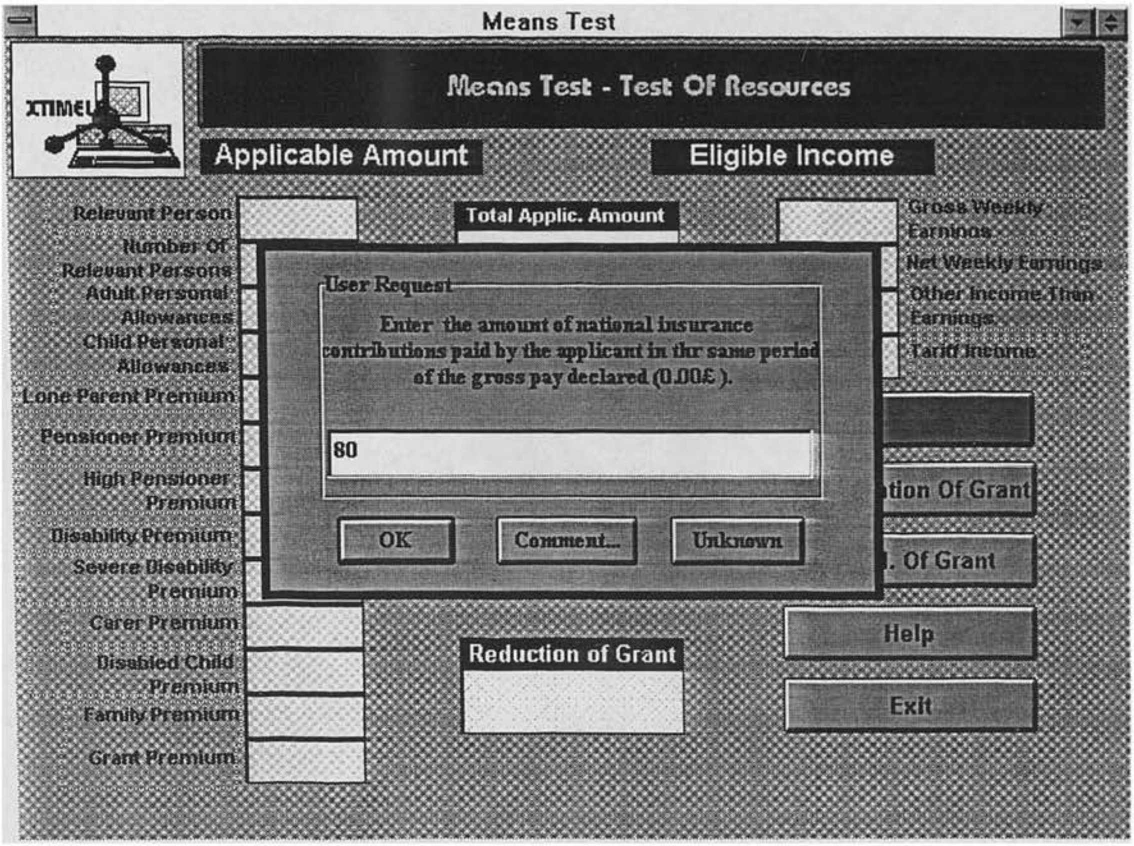
Screen 216: Input information for the test of resources



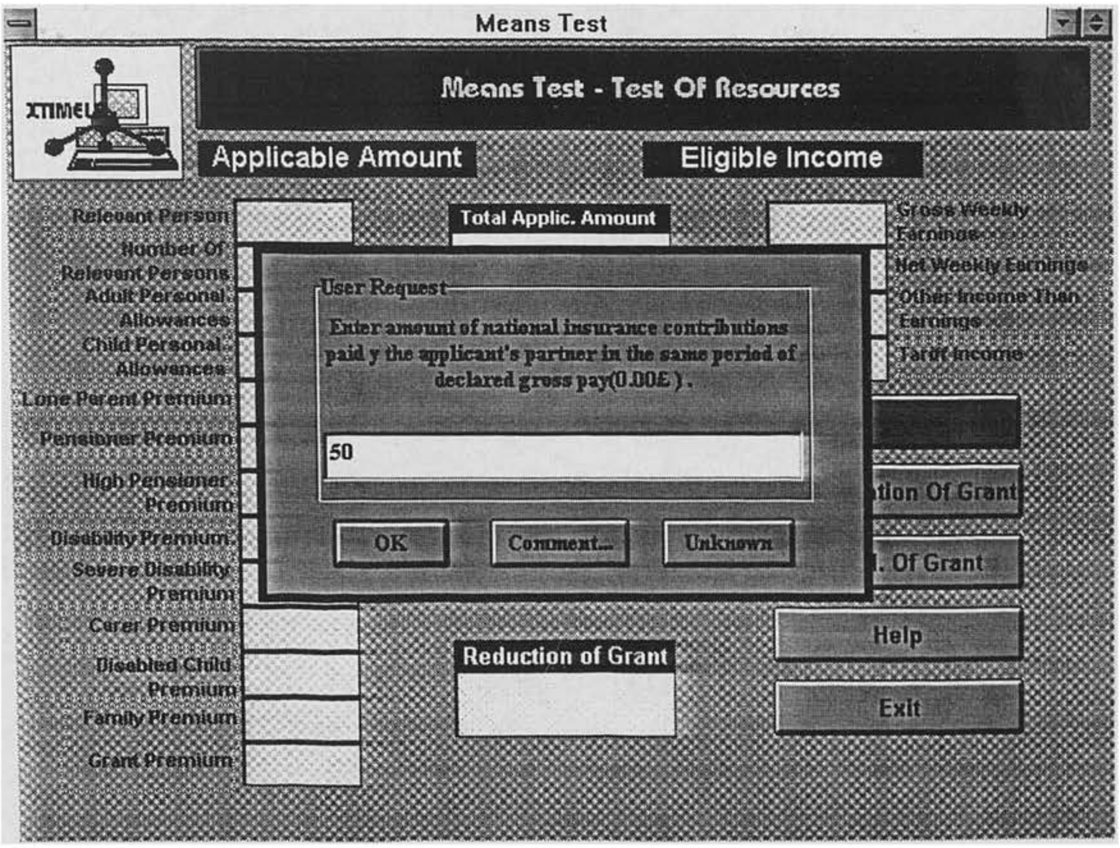
Screen 217: Input information for the test of resources



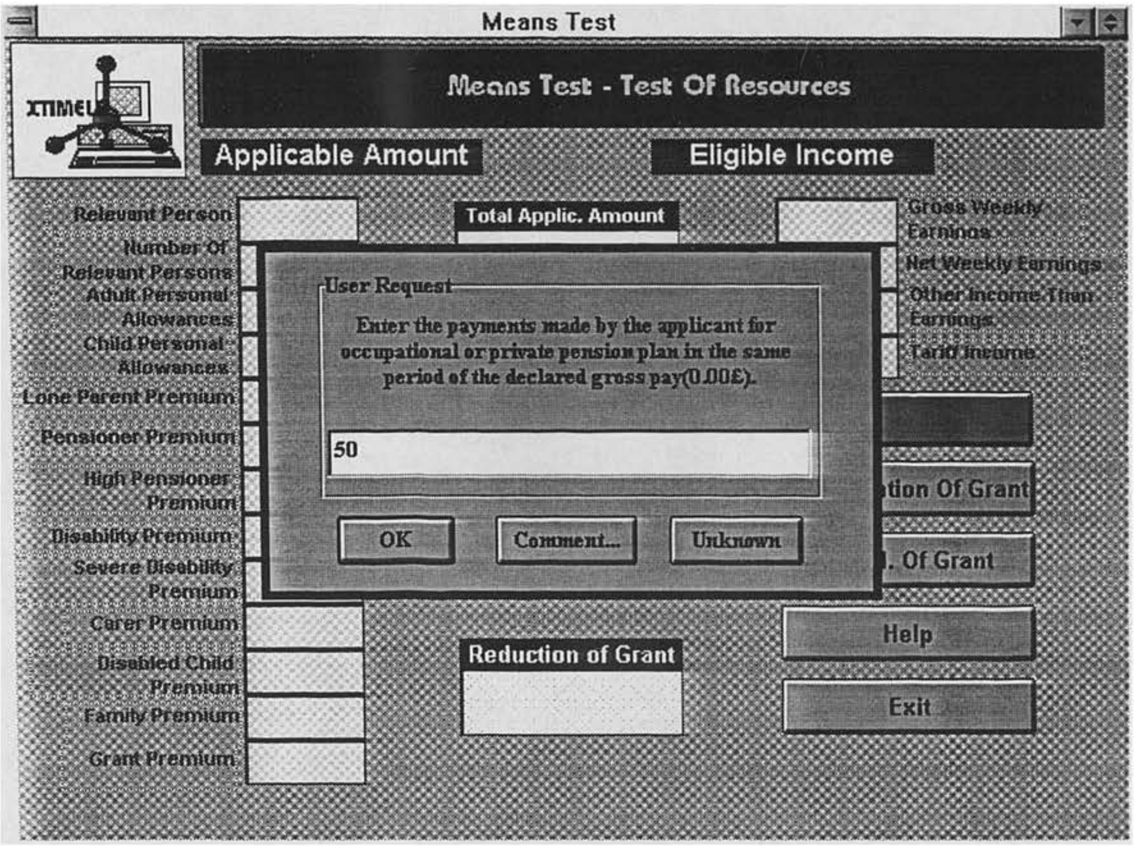
Screen 218: Input information for the test of resources



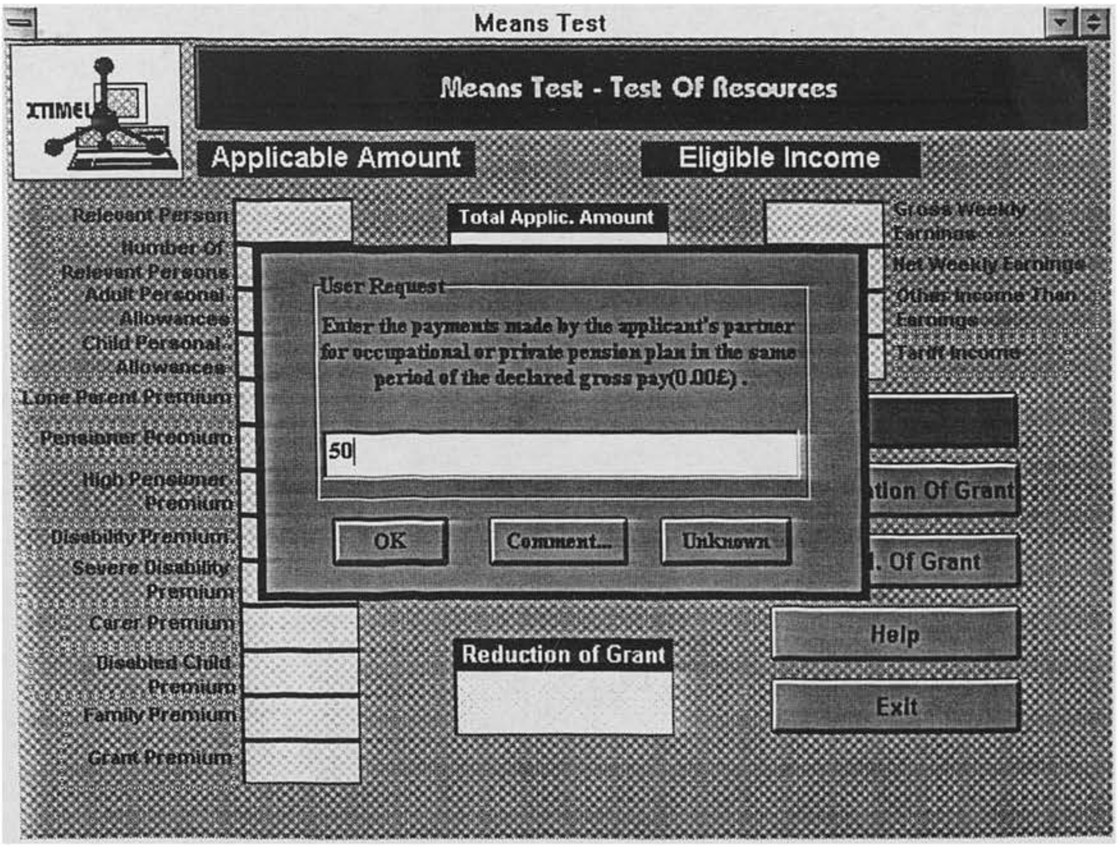
Screen 219: Input information for the test of resources



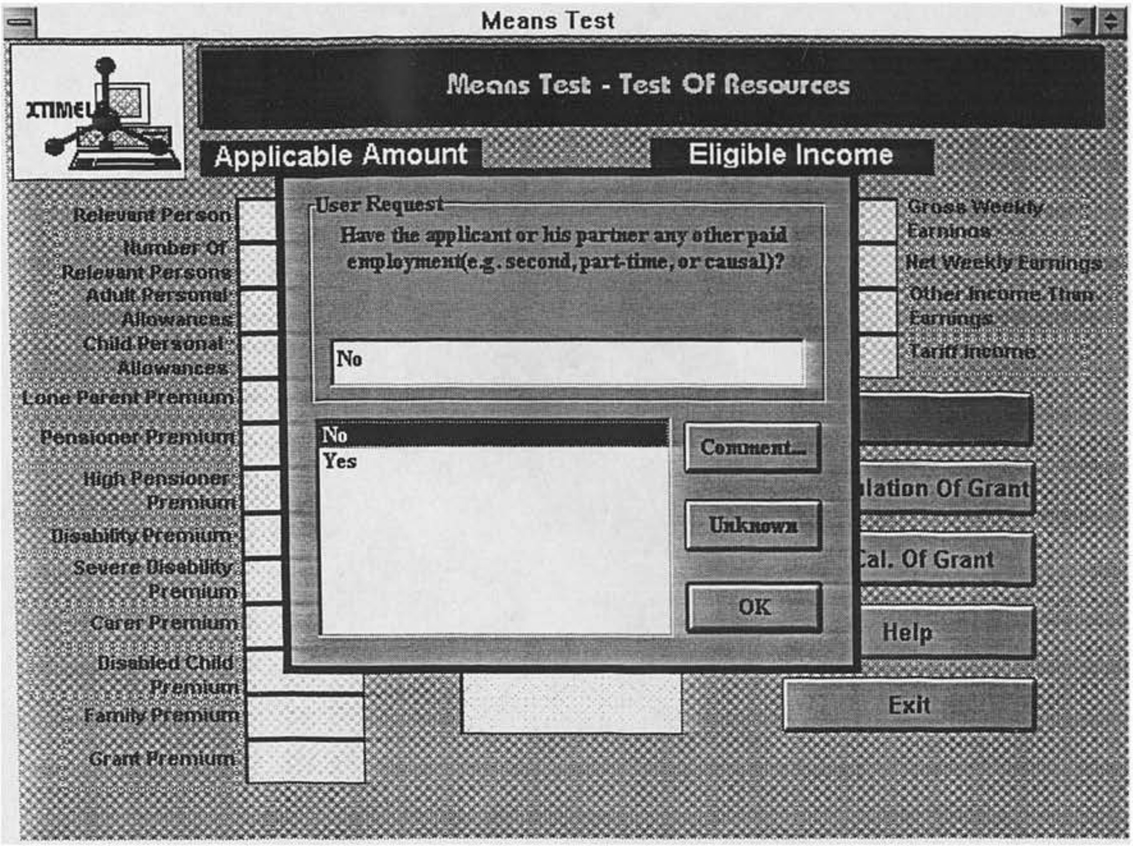
Screen 220: Input information for the test of resources



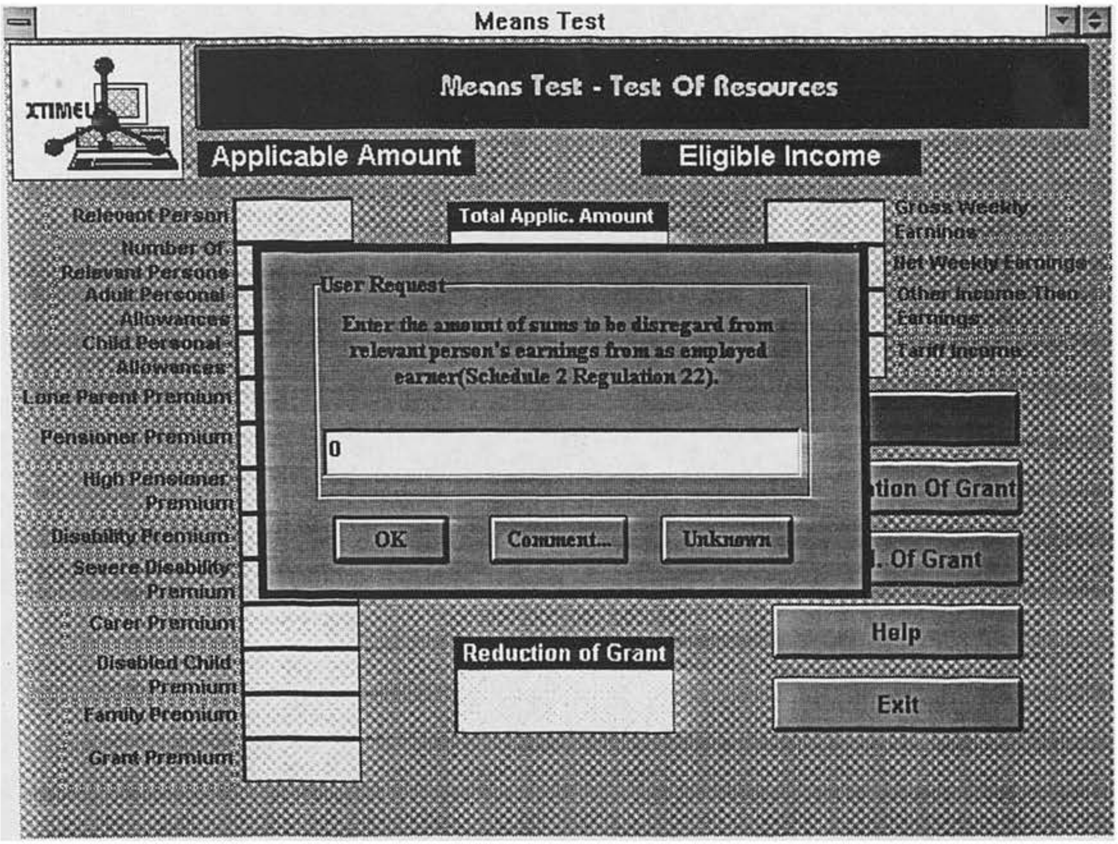
Screen 221: Input information for the test of resources



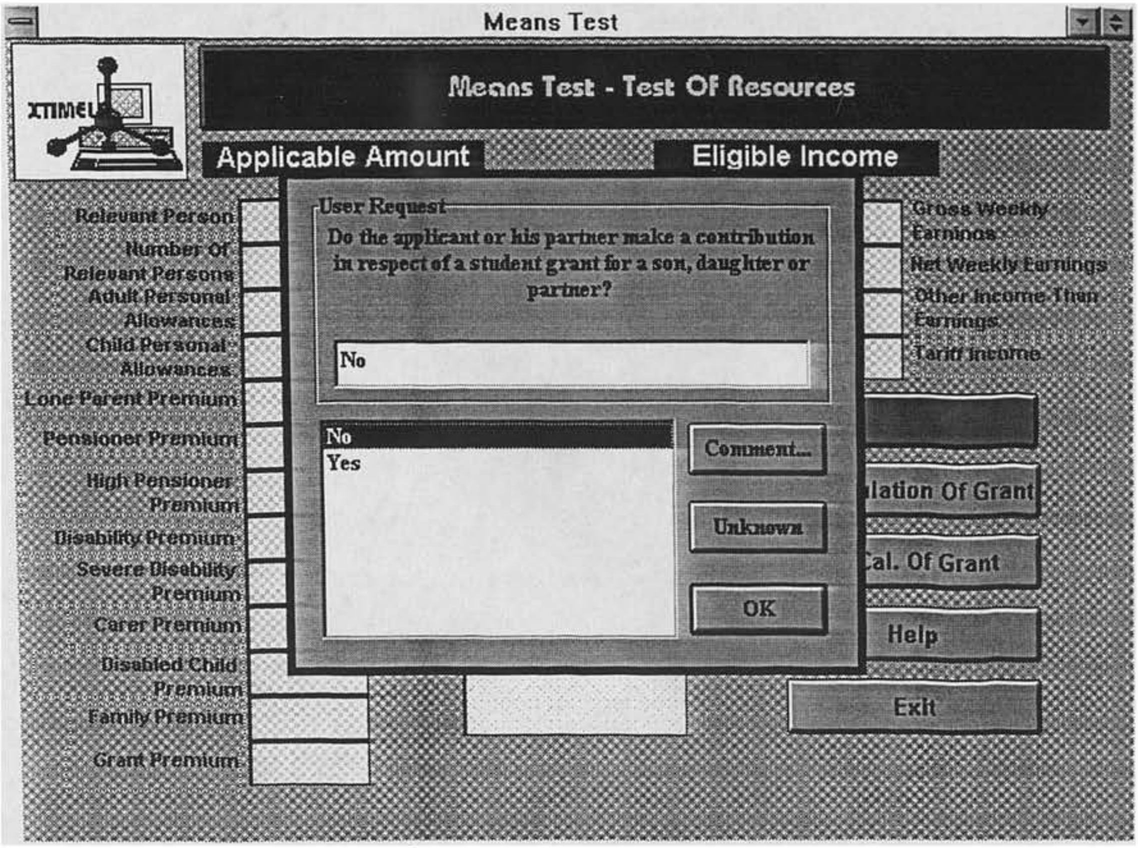
Screen 222: Input information for the test of resources



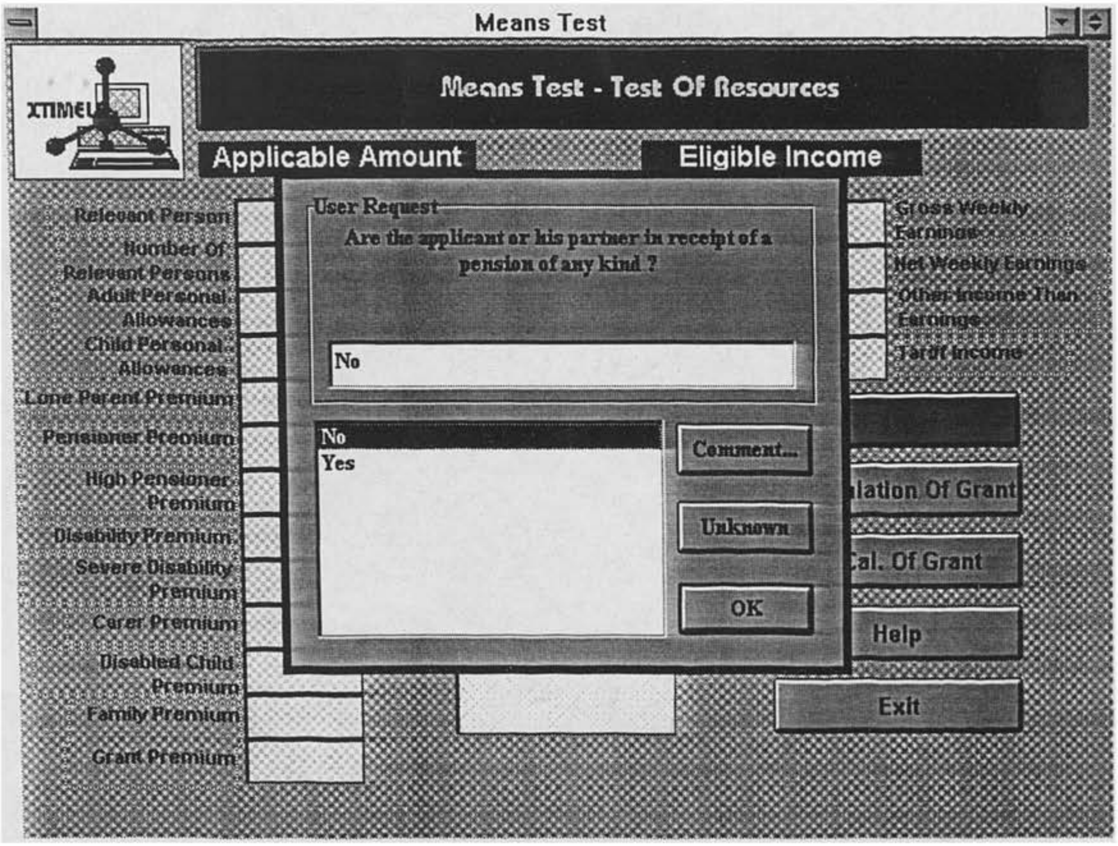
Screen 223: Input information for the test of resources



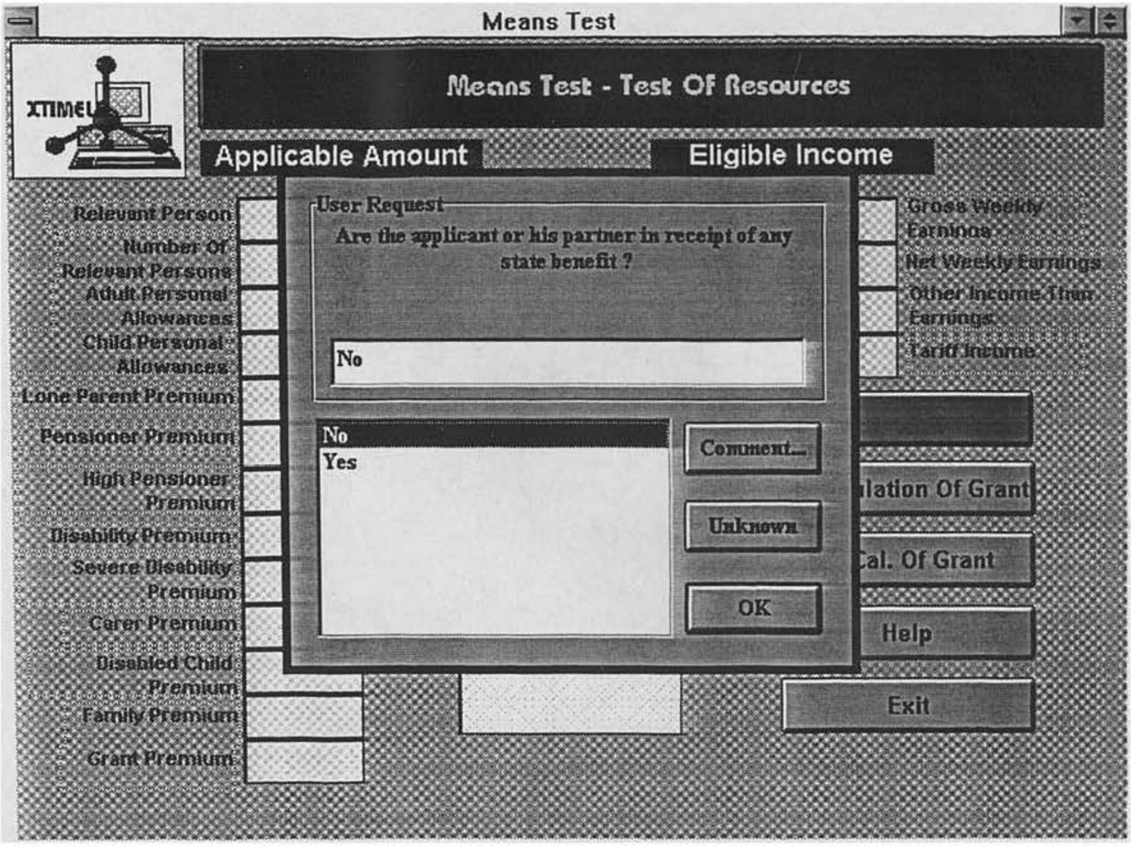
Screen 224: Input information for the test of resources



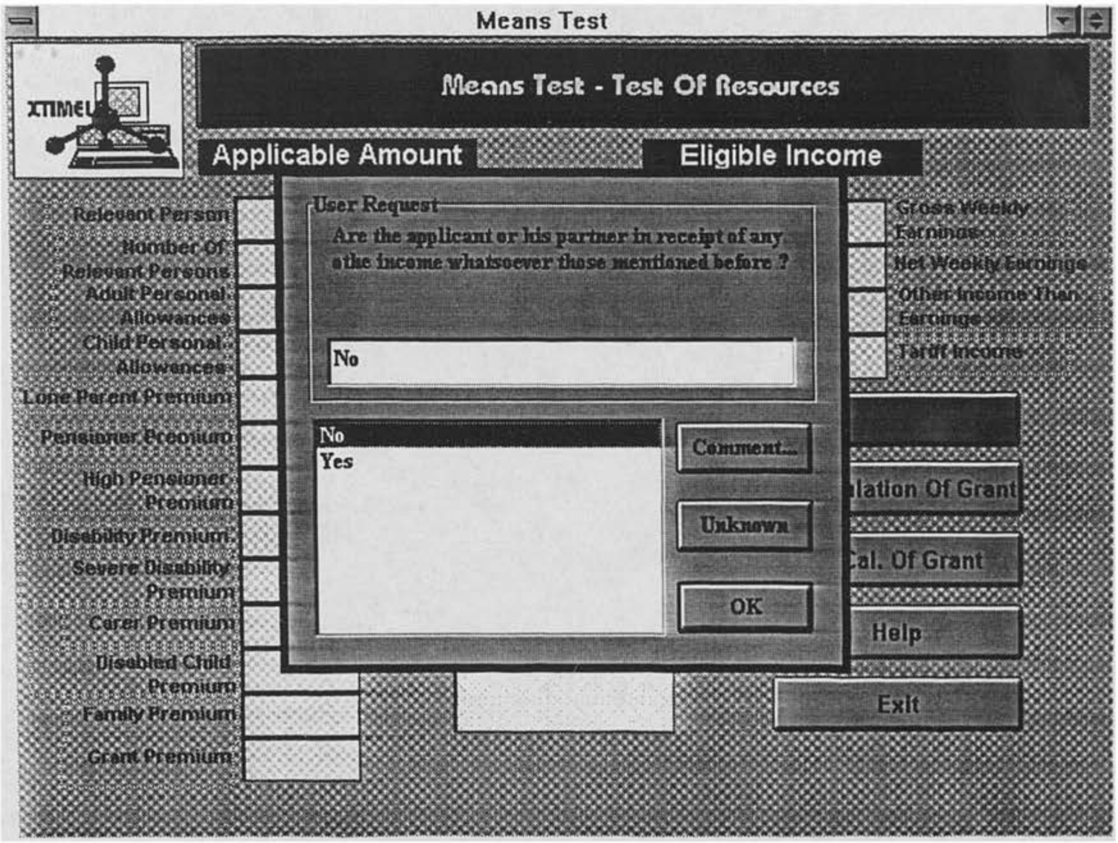
Screen 225: Input information for the test of resources



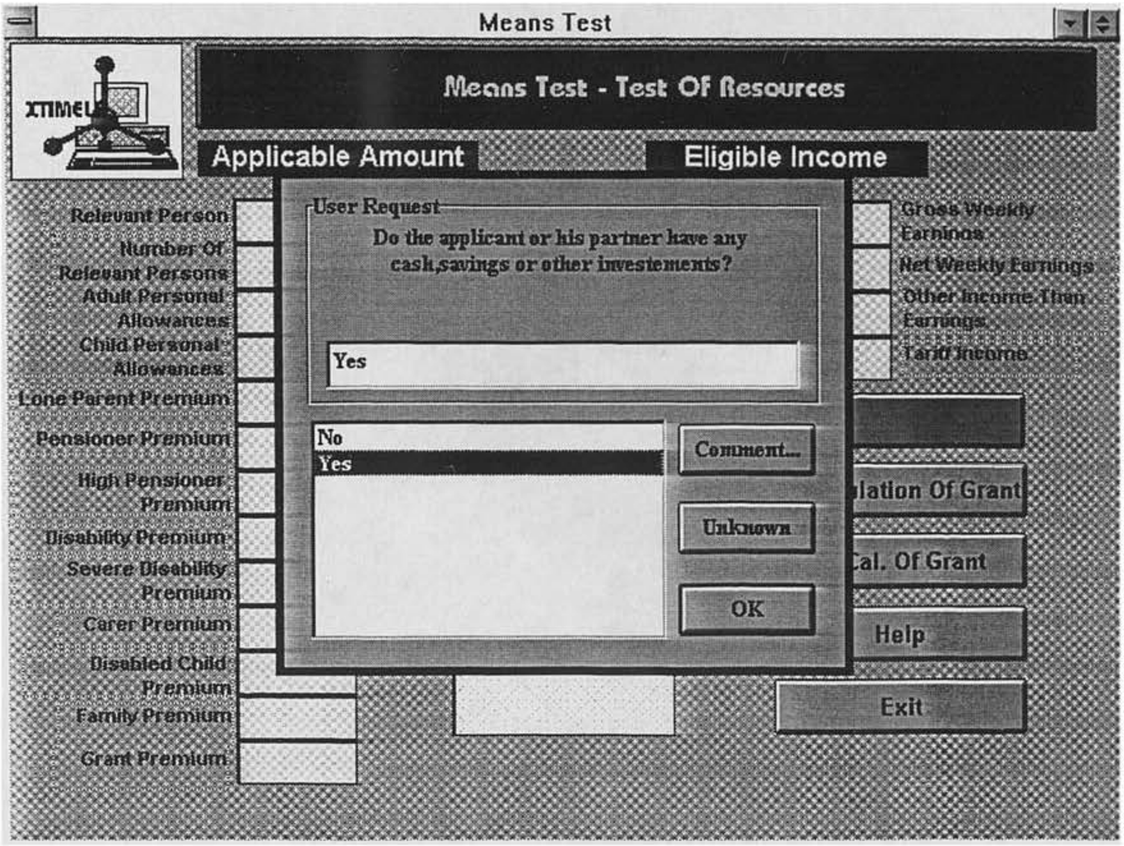
Screen 226: Input information for the test of resources



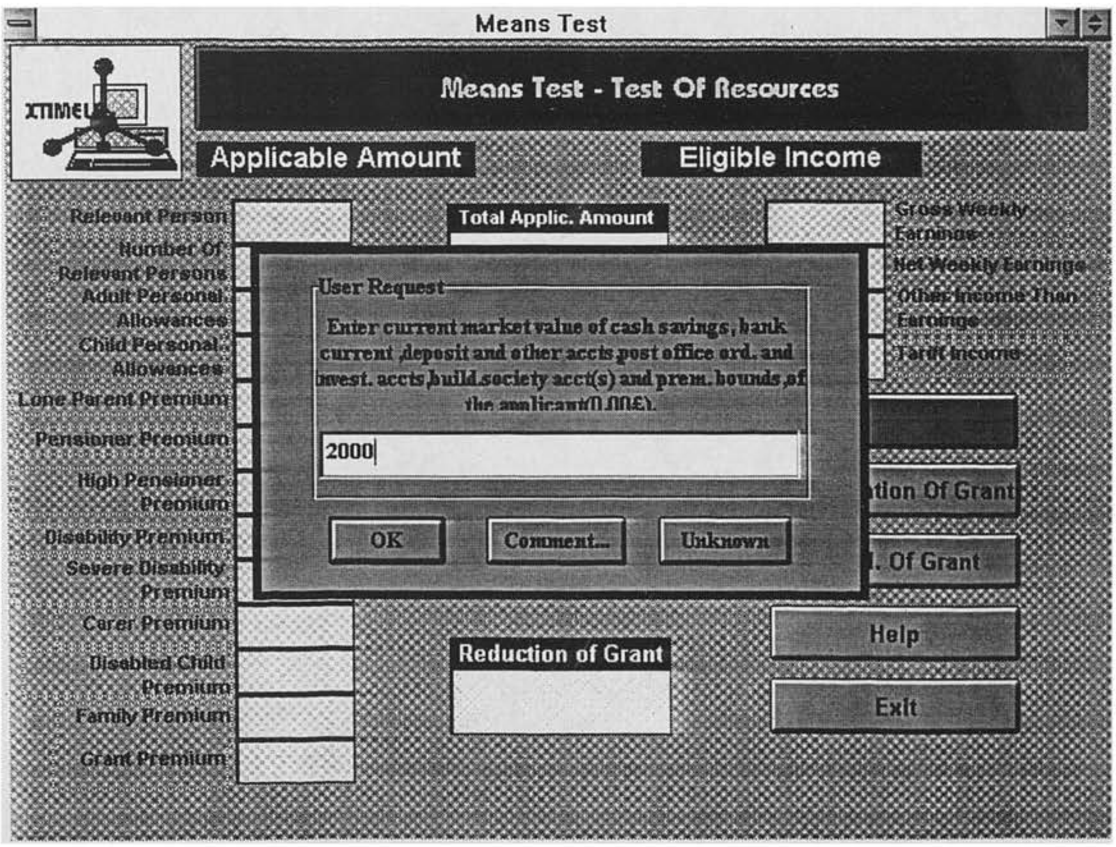
Screen 227: Input information for the test of resources



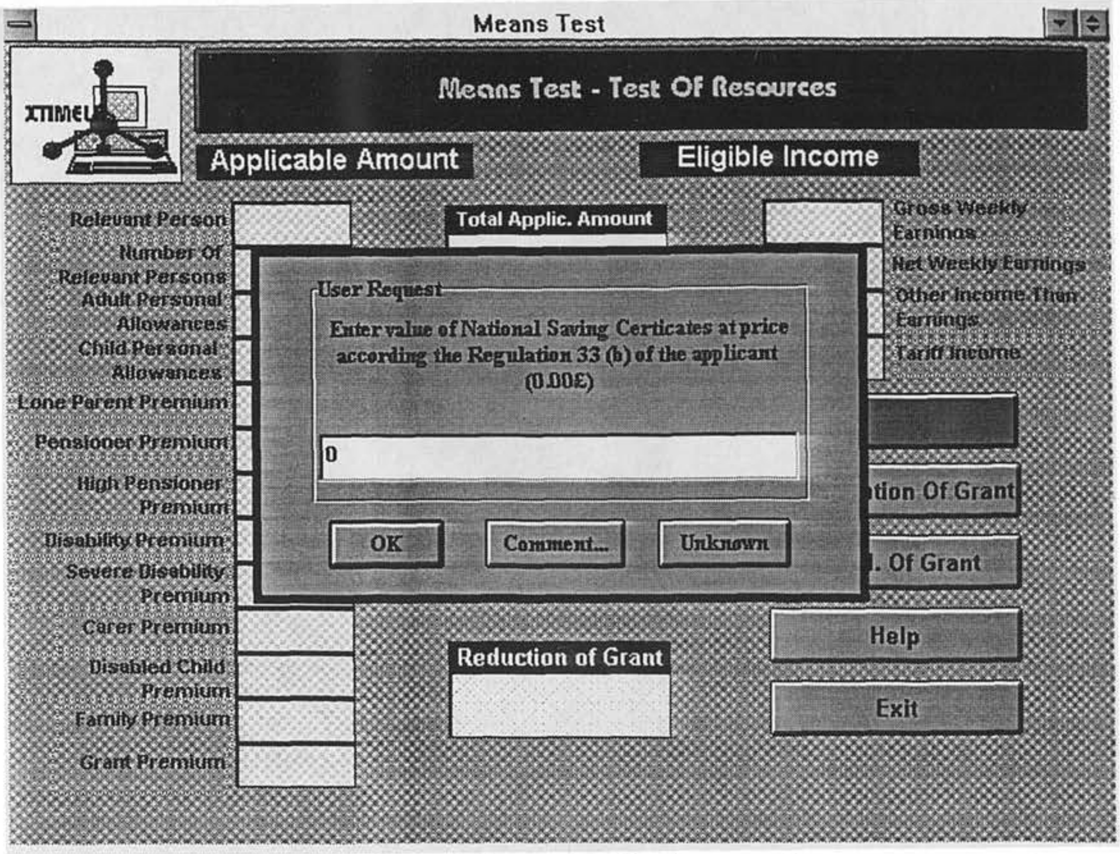
Screen 228: Input information for the test of resources



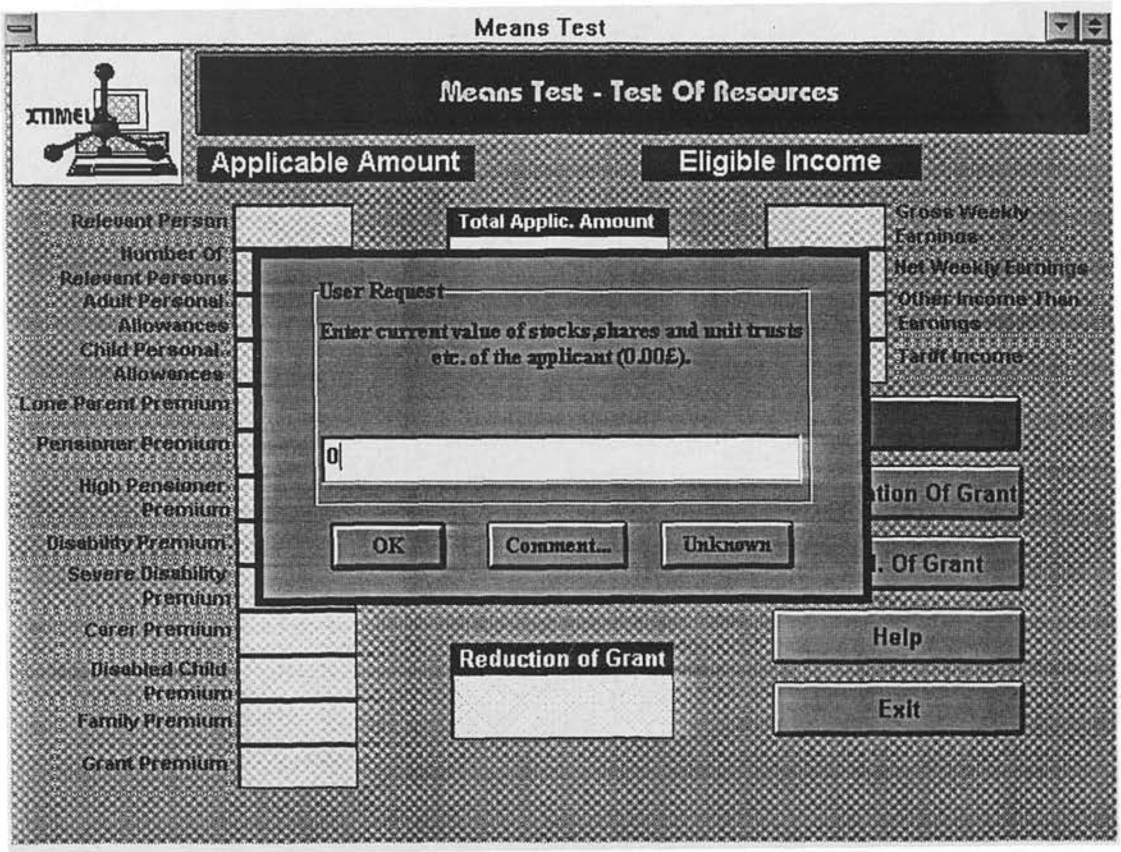
Screen 229: Input information for the test of resources



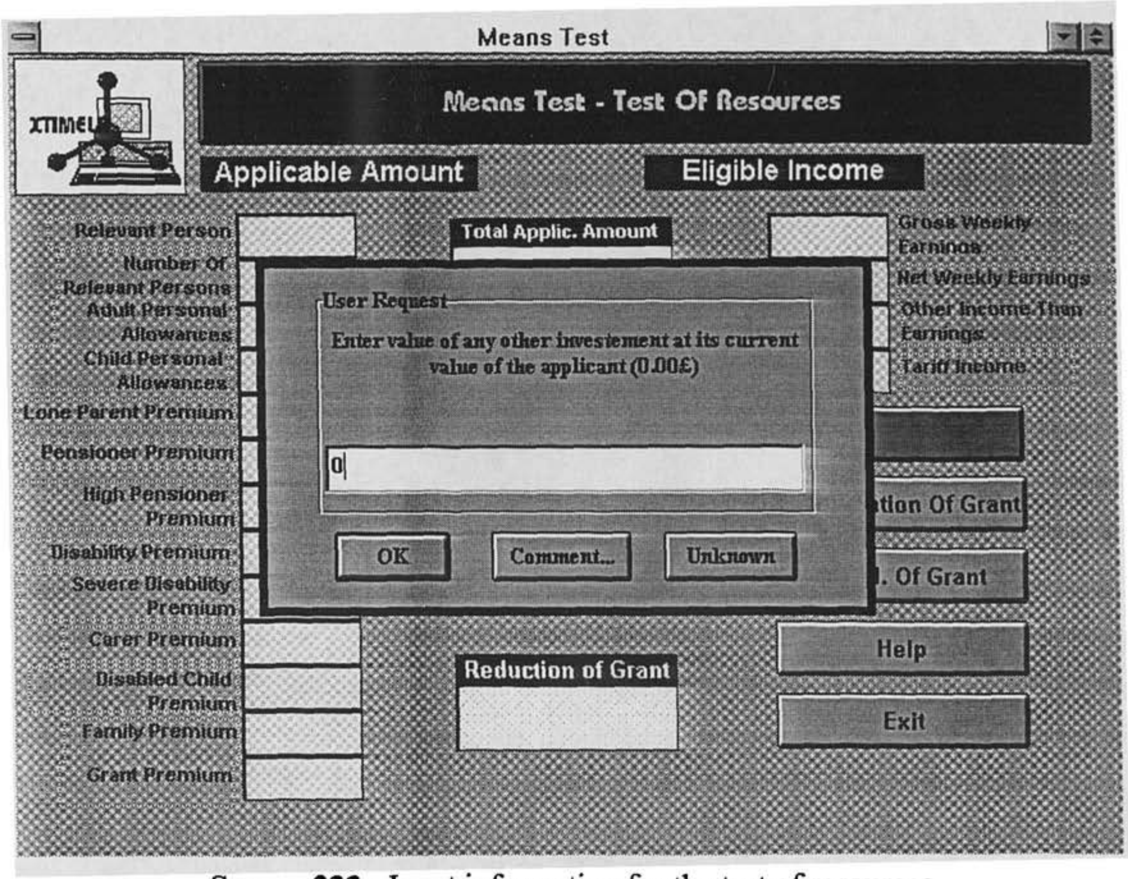
Screen 230: Input information for the test of resources



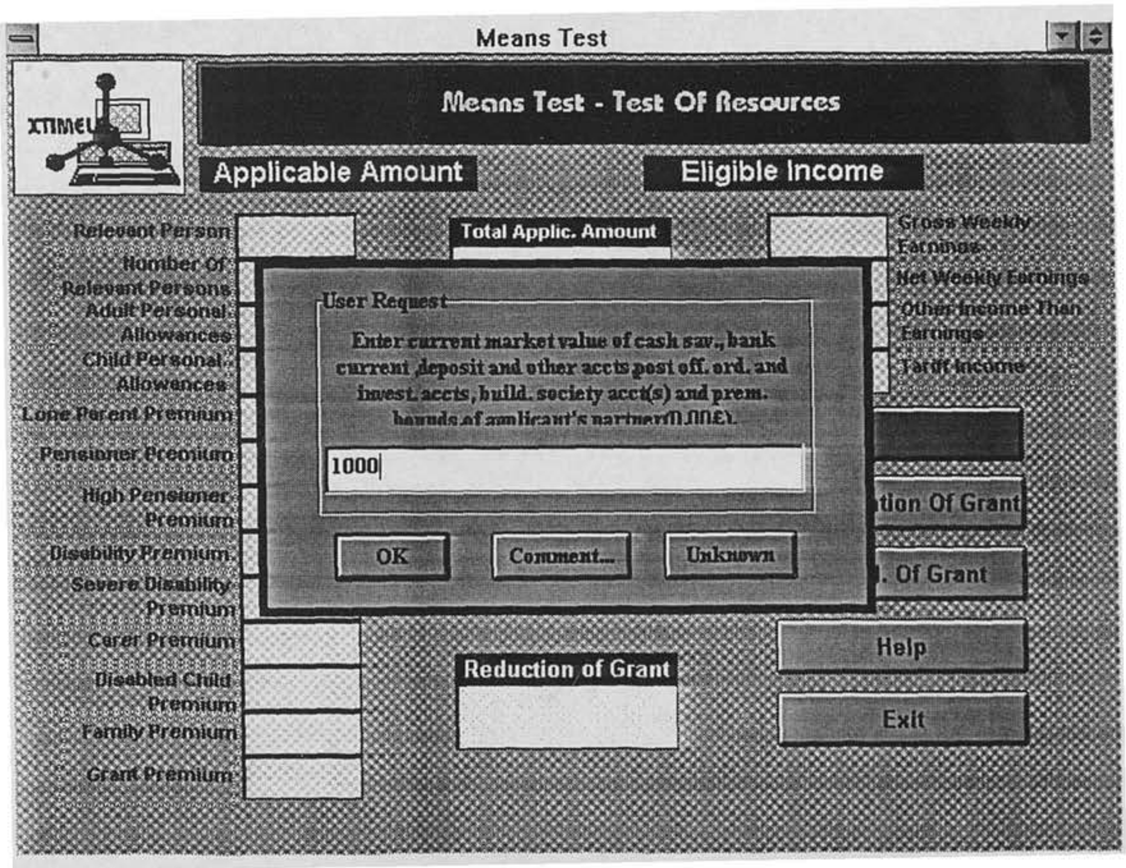
Screen 231: Input information for the test of resources



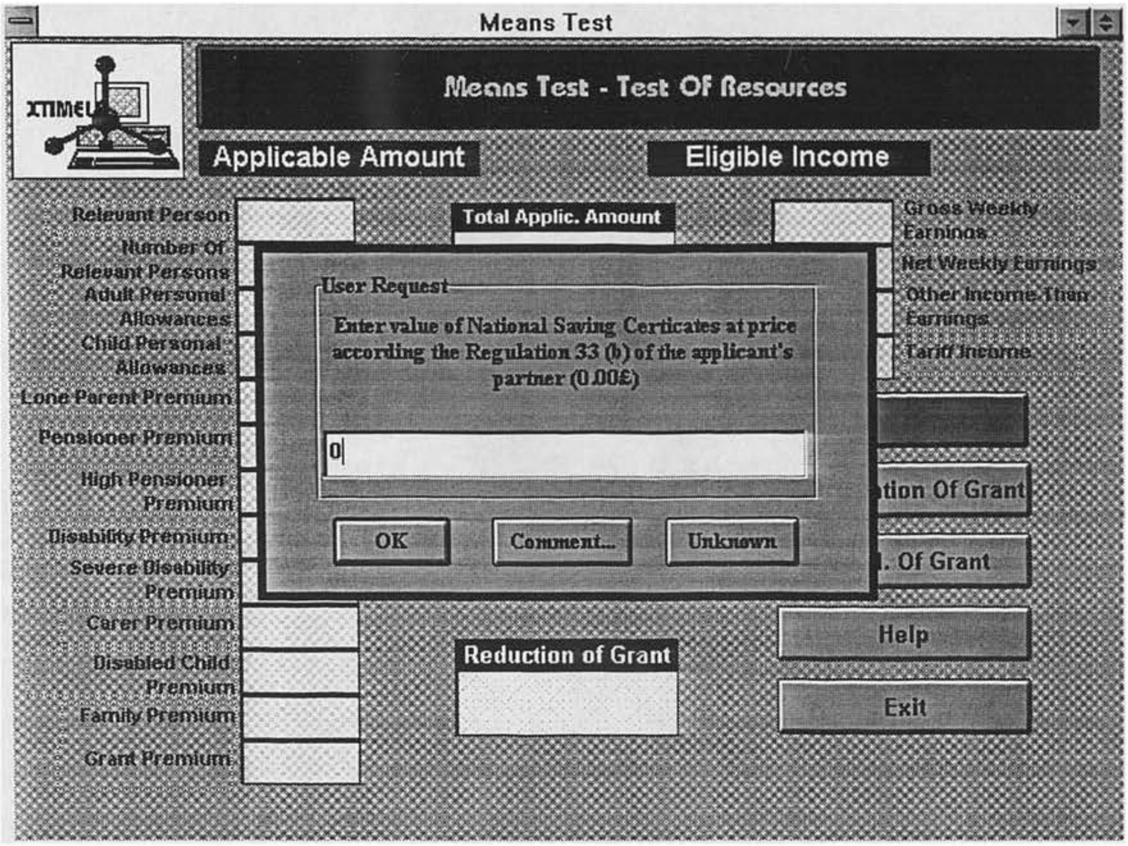
Screen 232: Input information for the test of resources



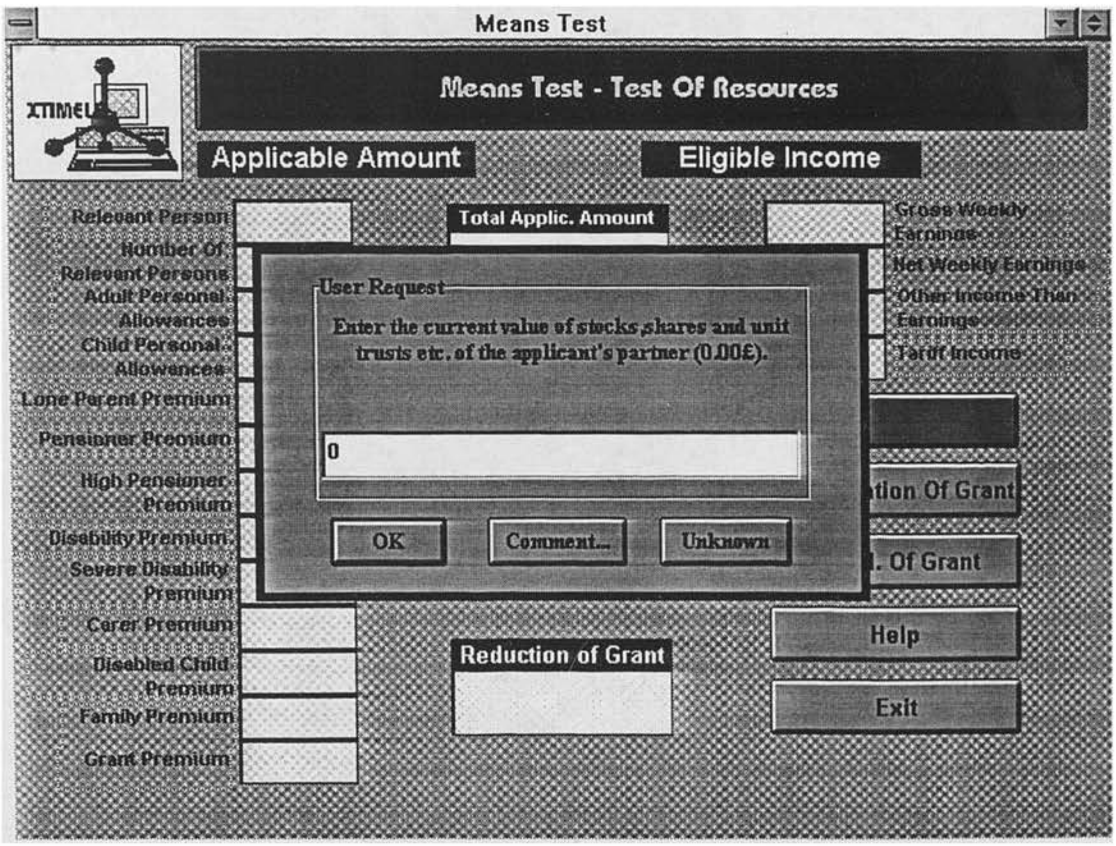
Screen 233 : Input information for the test of resources



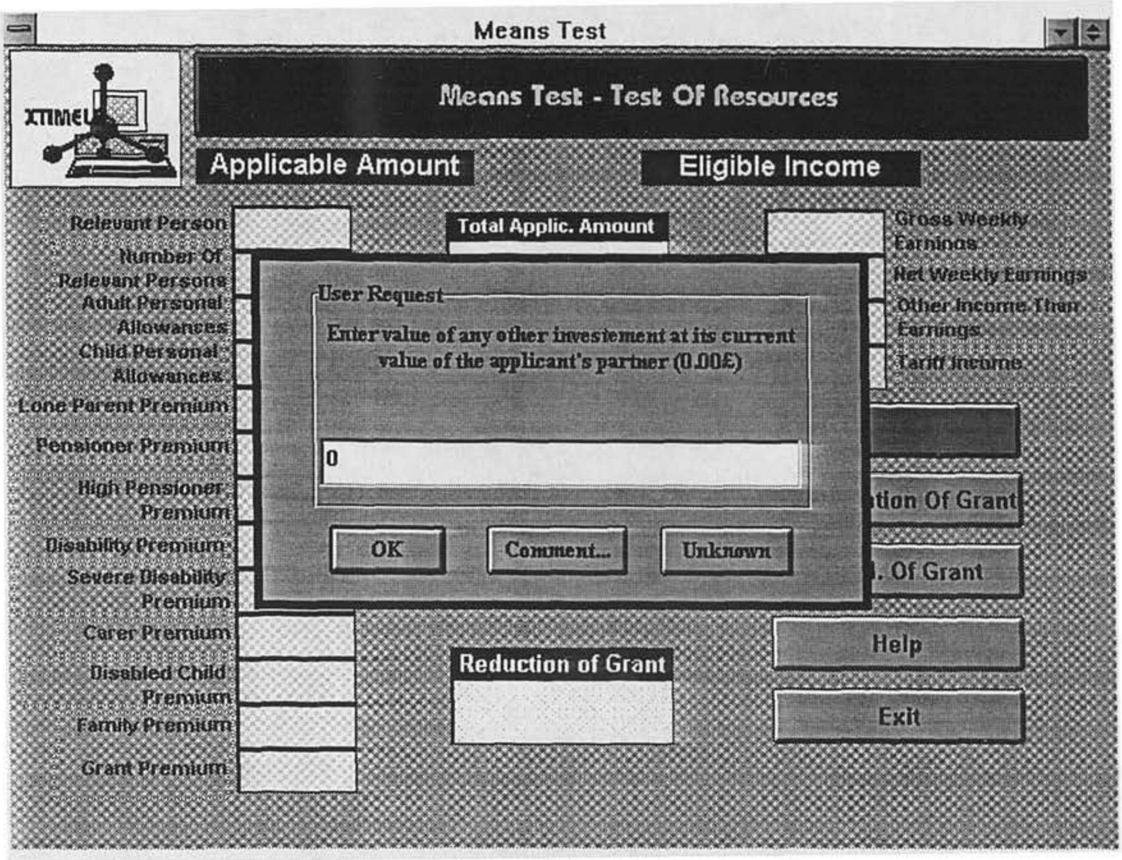
Screen 234: Input information for the test of resources



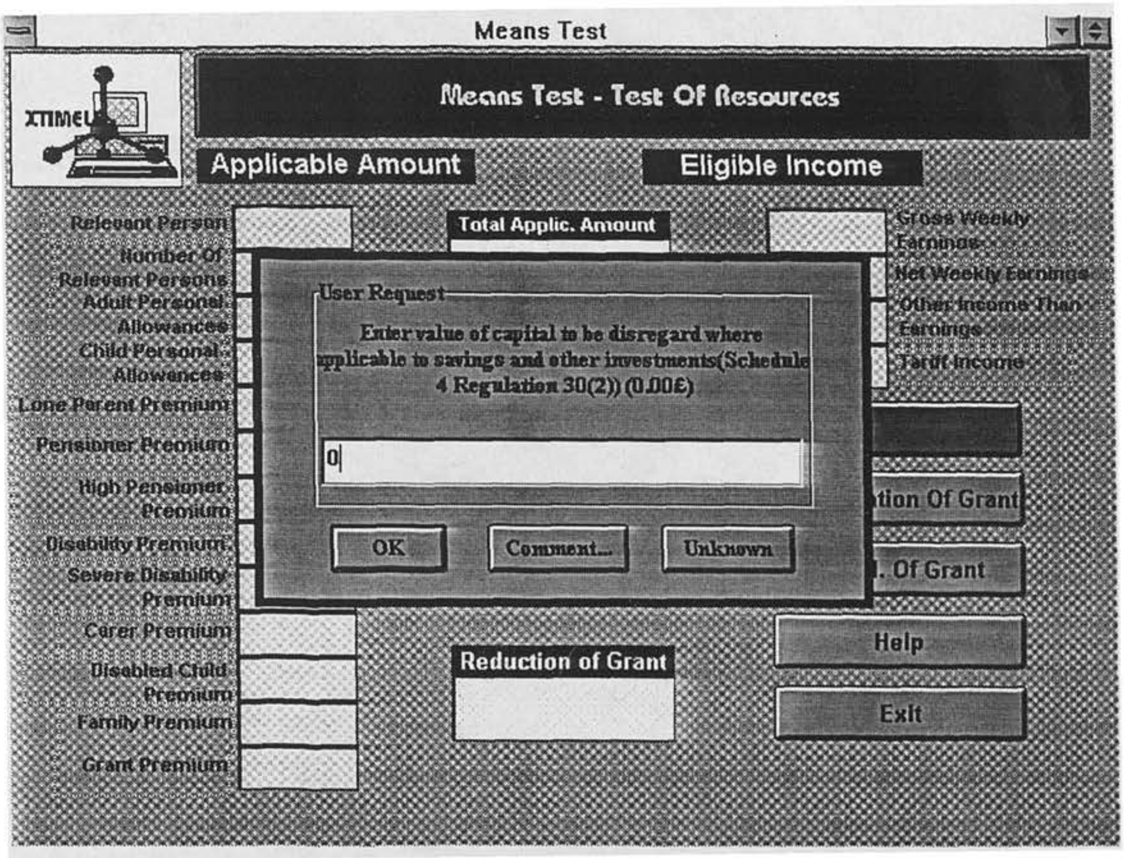
Screen 235: Input information for the test of resources



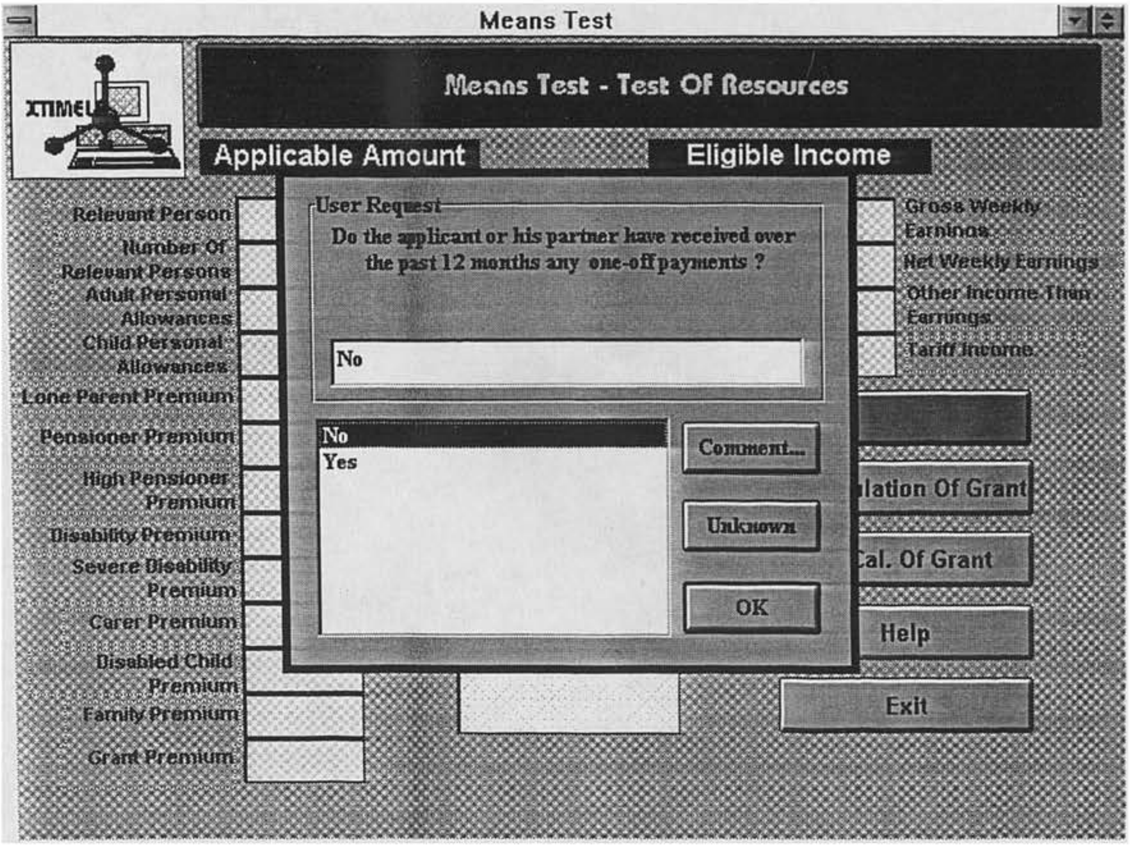
Screen 236: Input information for the test of resources



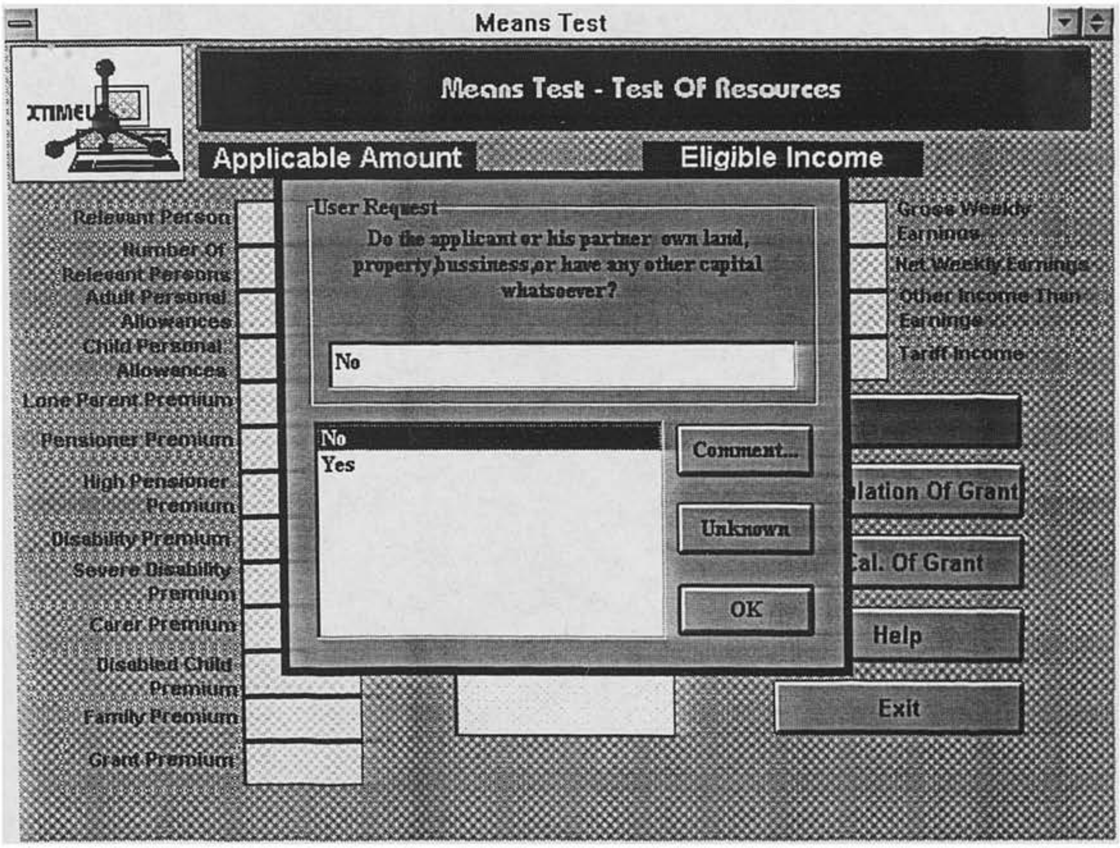
Screen 237: Input information for the test of resources



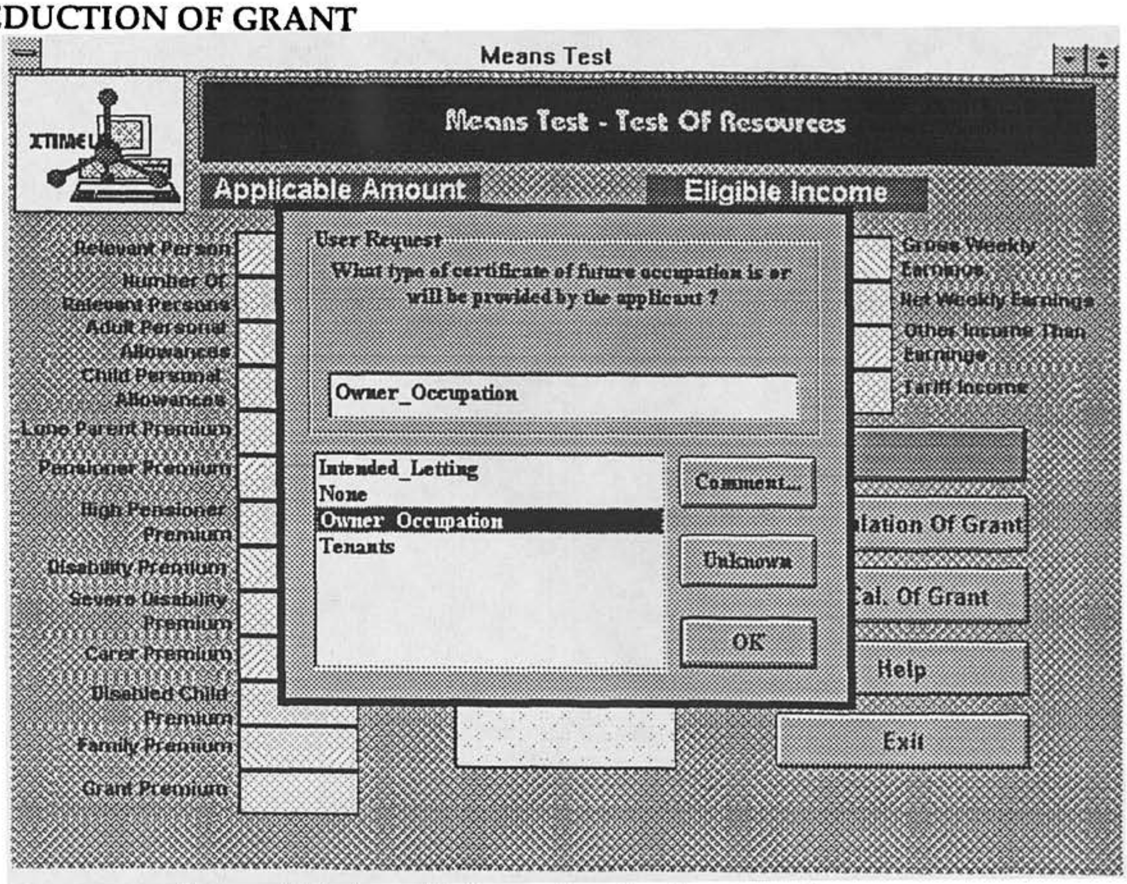
Screen 238: Input information for the test of resources



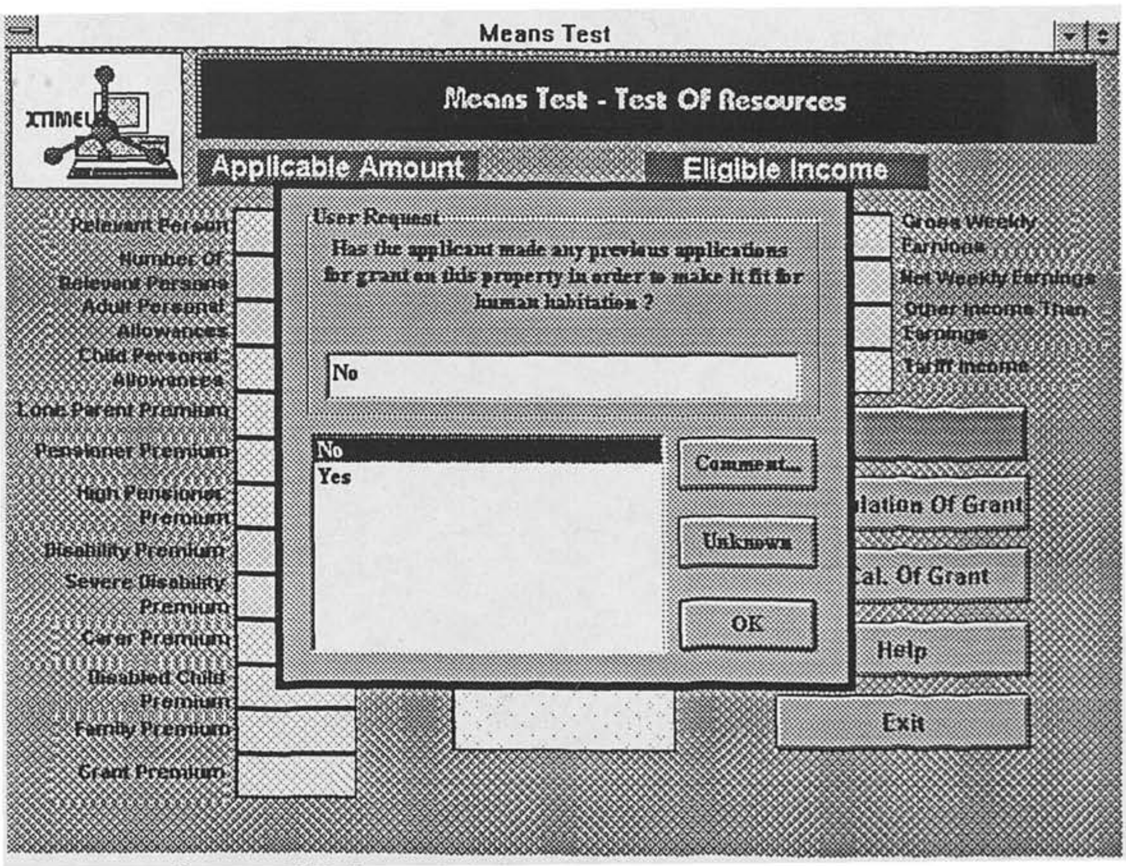
Screen 239: Input information for the test of resources



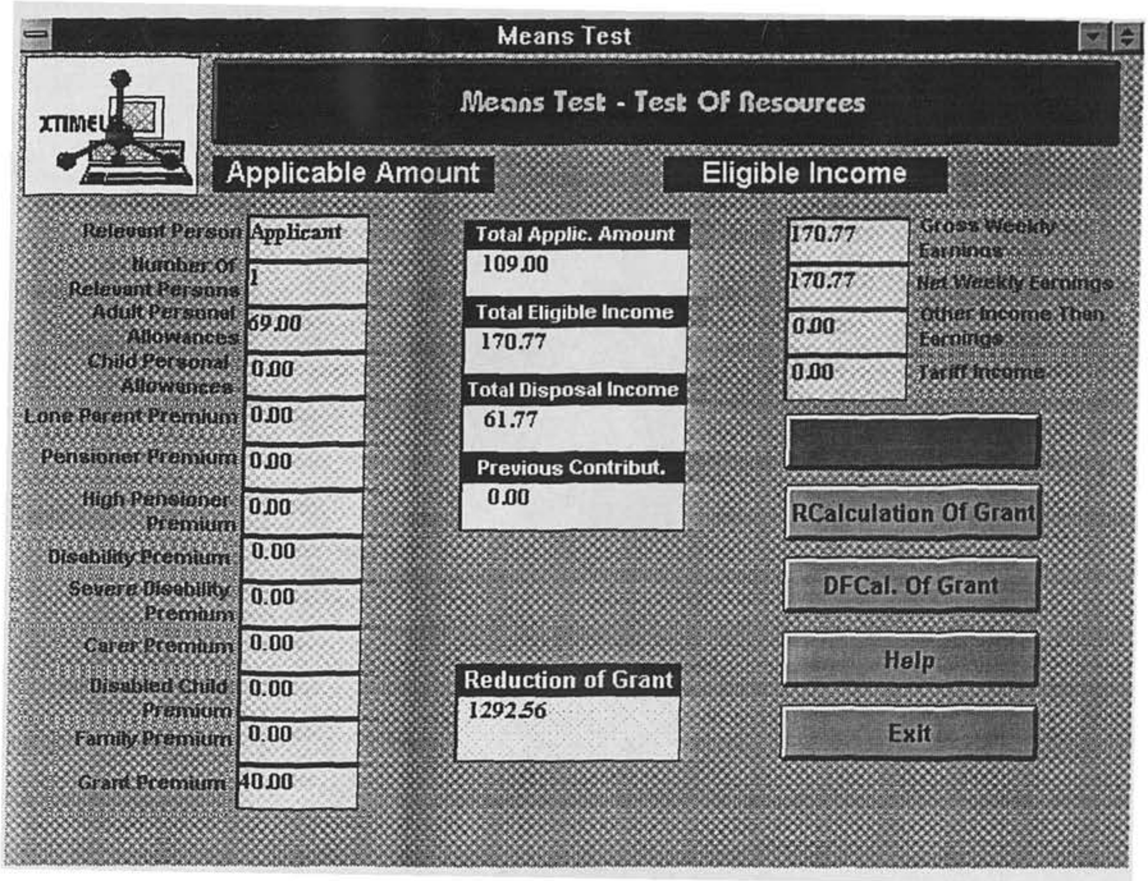
Screen 240: Input information for the test of resources



Screen 241: Input information for the test of resources

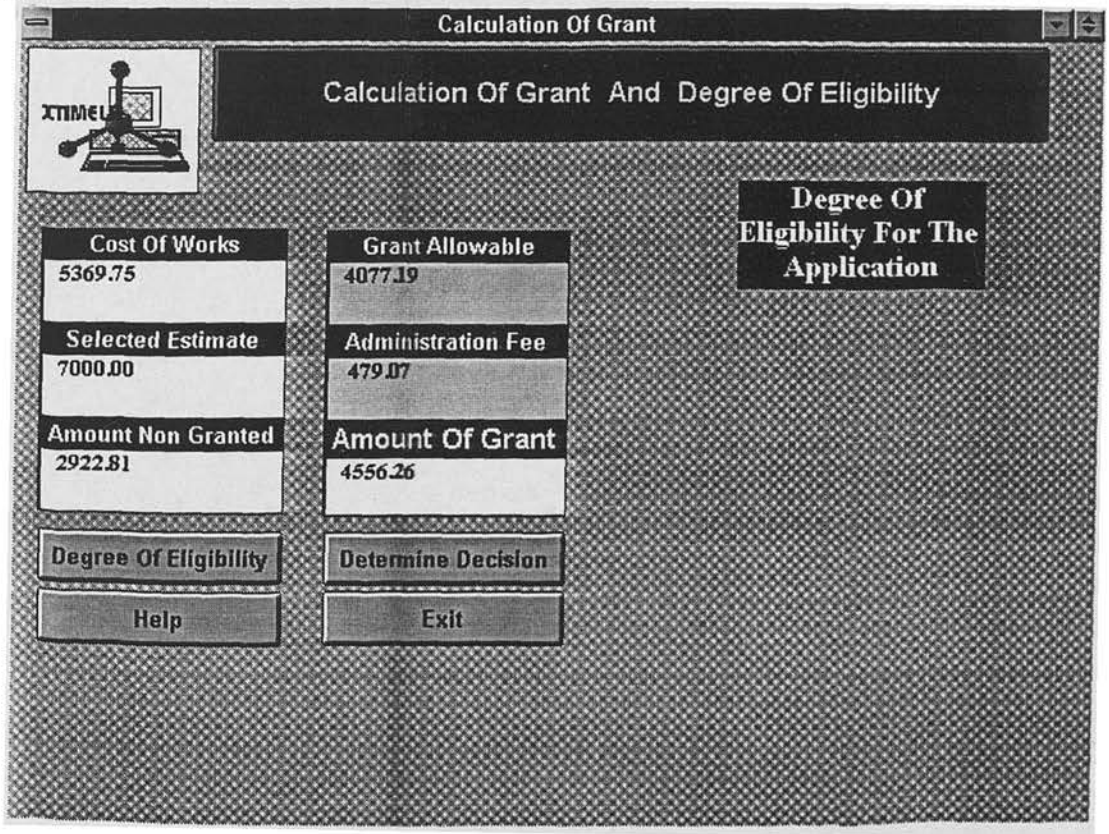


Screen 242: Input information for the test of resources



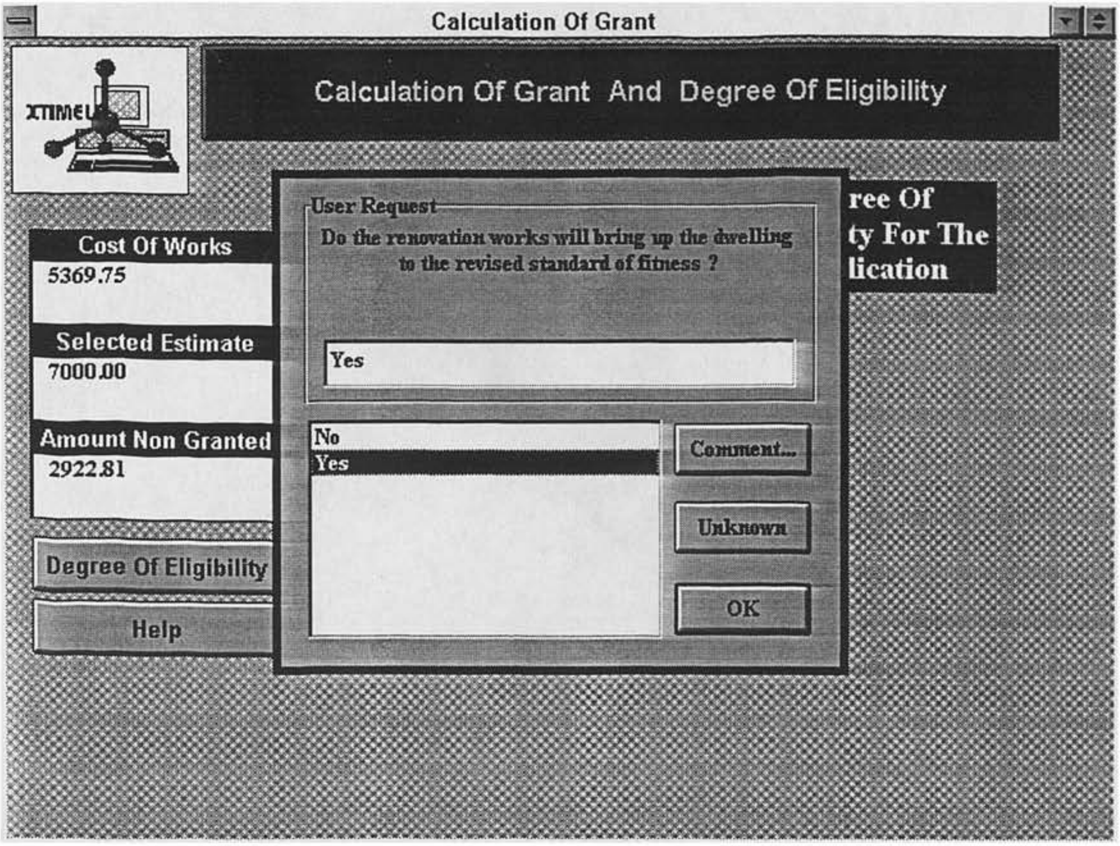
Screen 243: Report of the means test results

10.9- AMOUNT OF GRANT

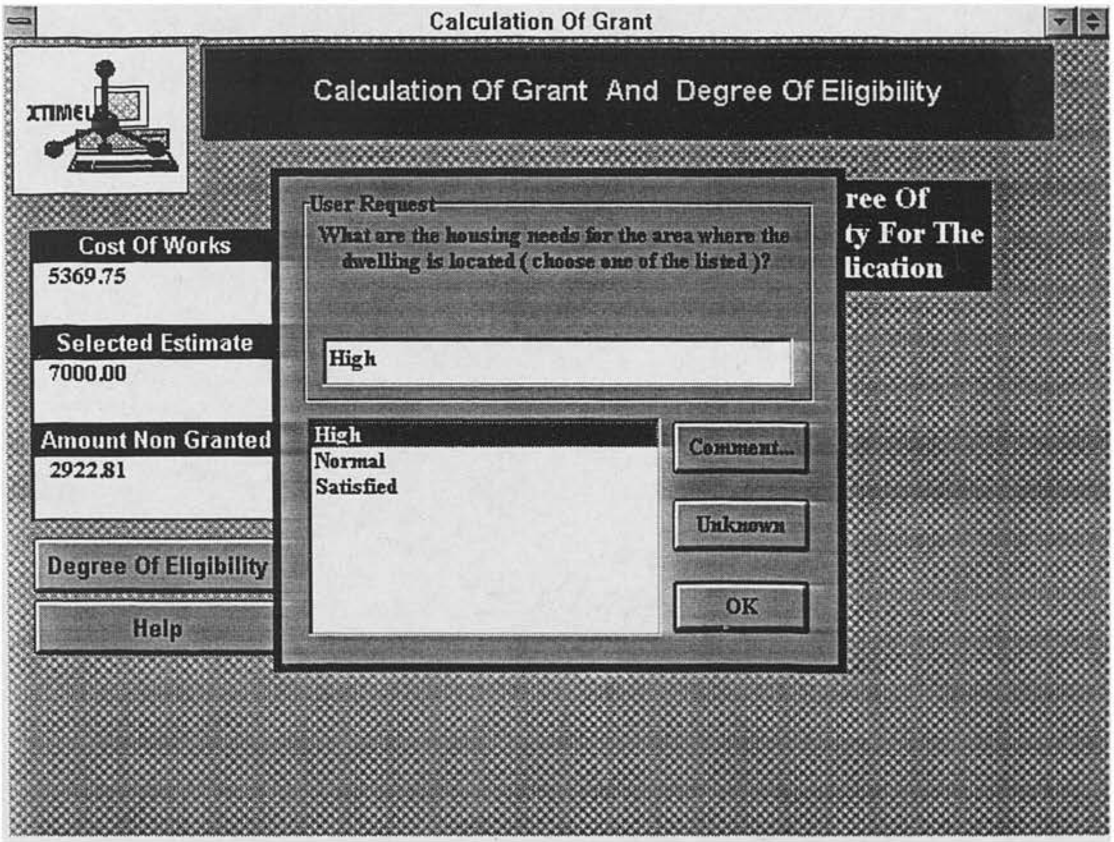


Screen 244: Report of the amount of grant

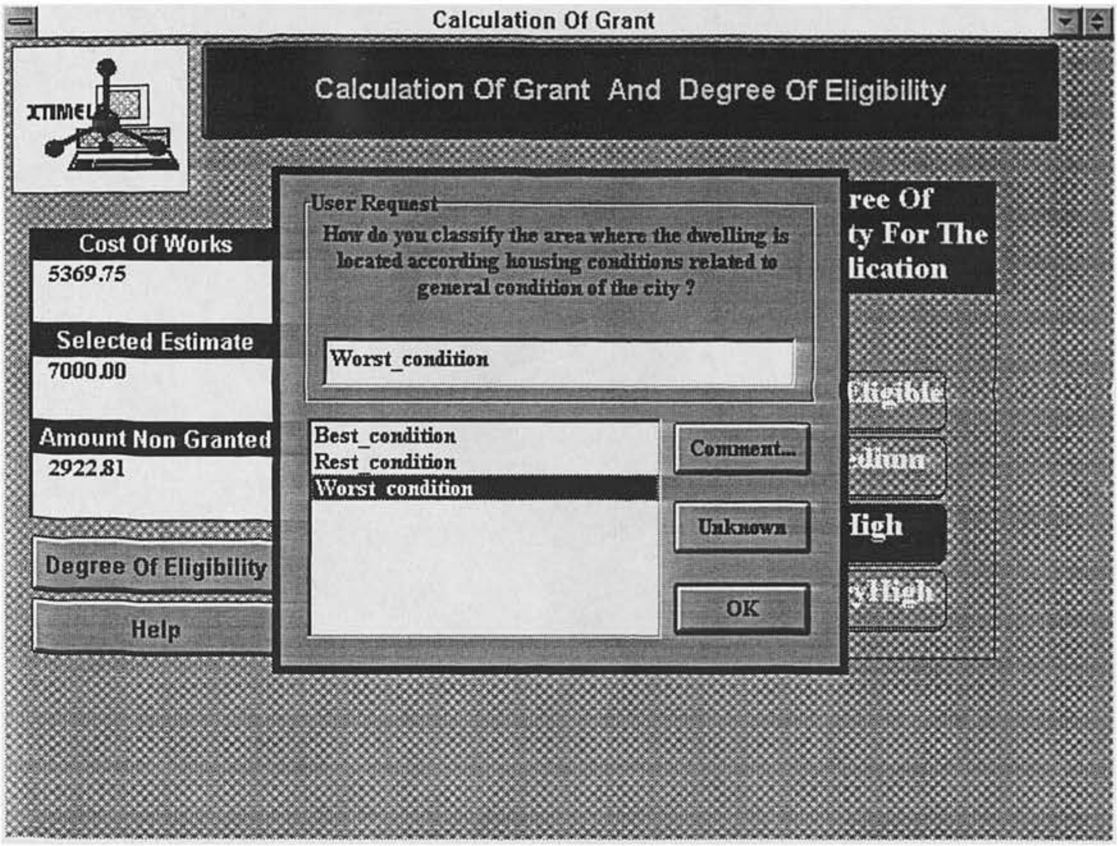
10.10- DEGREE OF ELIGIBILITY



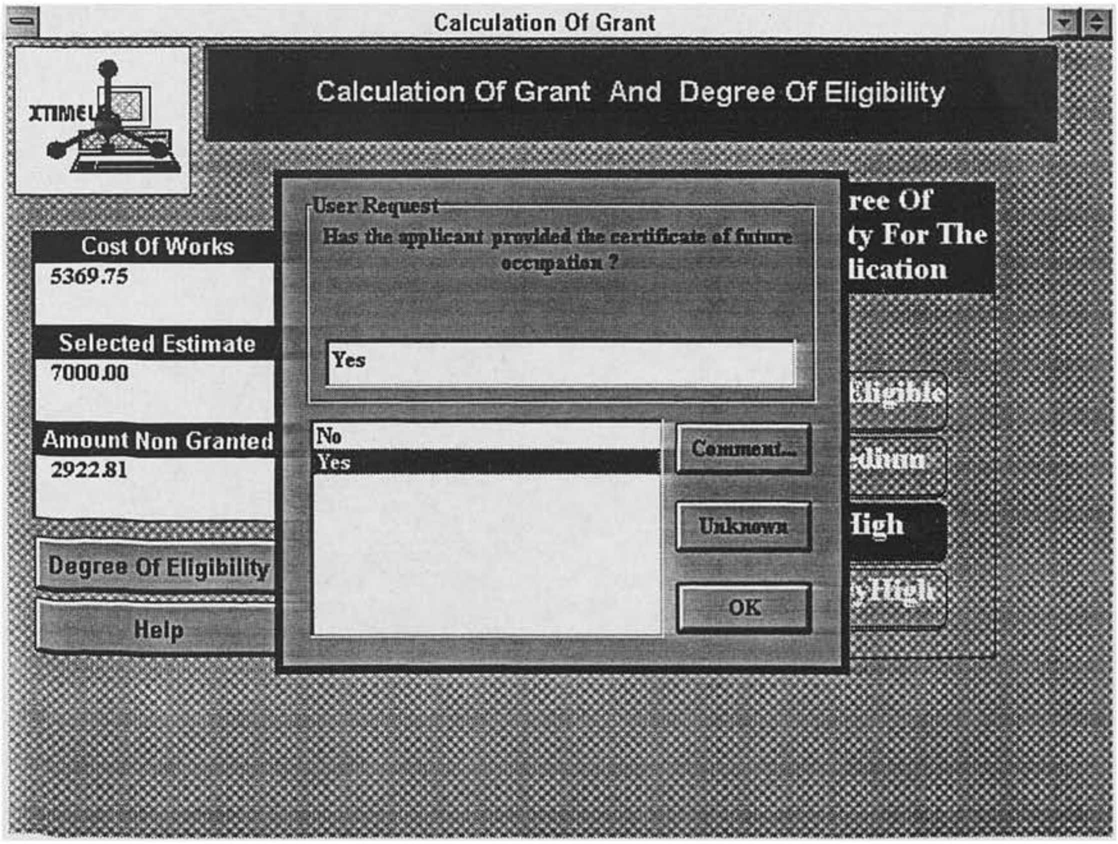
Screen 245: Input information for determing the degree of eligibility



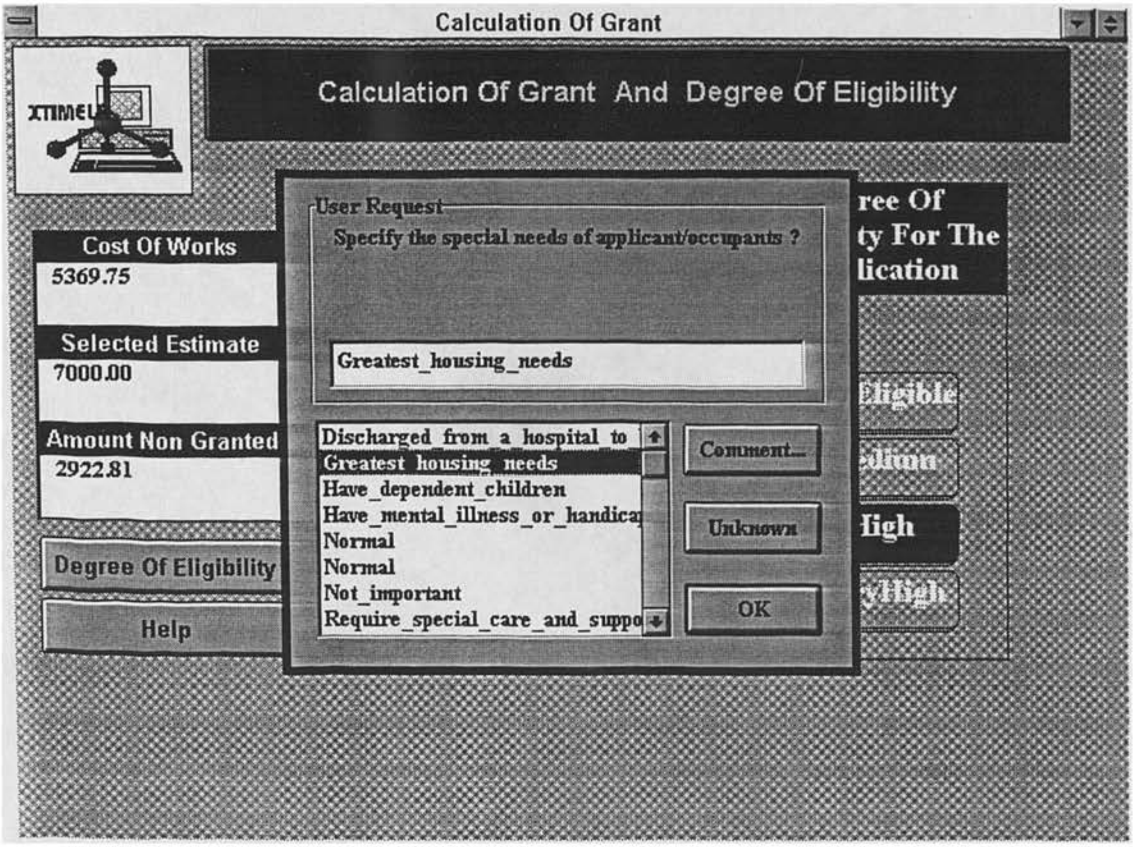
Screen 246: Input information for determing the degree of eligibility



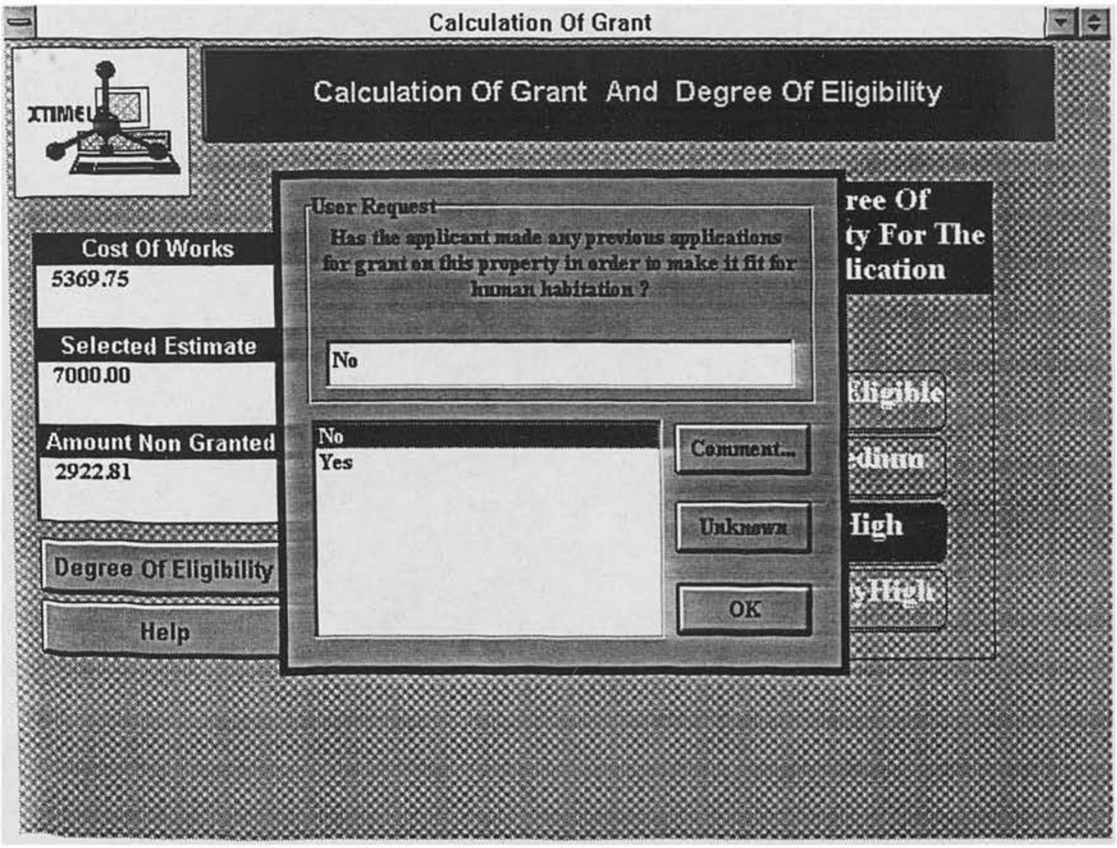
Screen 247: Input information for determing the degree of eligibility



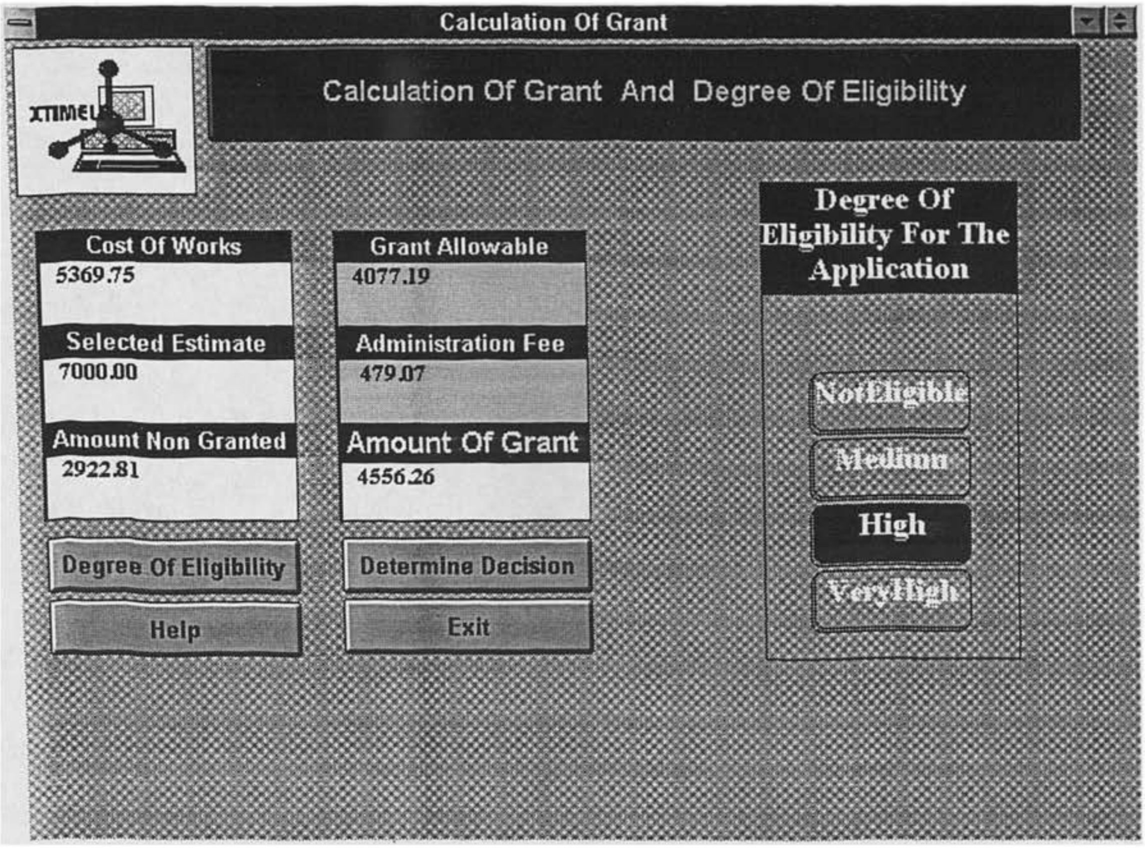
Screen 248: Input information for determing the degree of eligibility



Screen 249 : Input information for determing the degree of eligibility

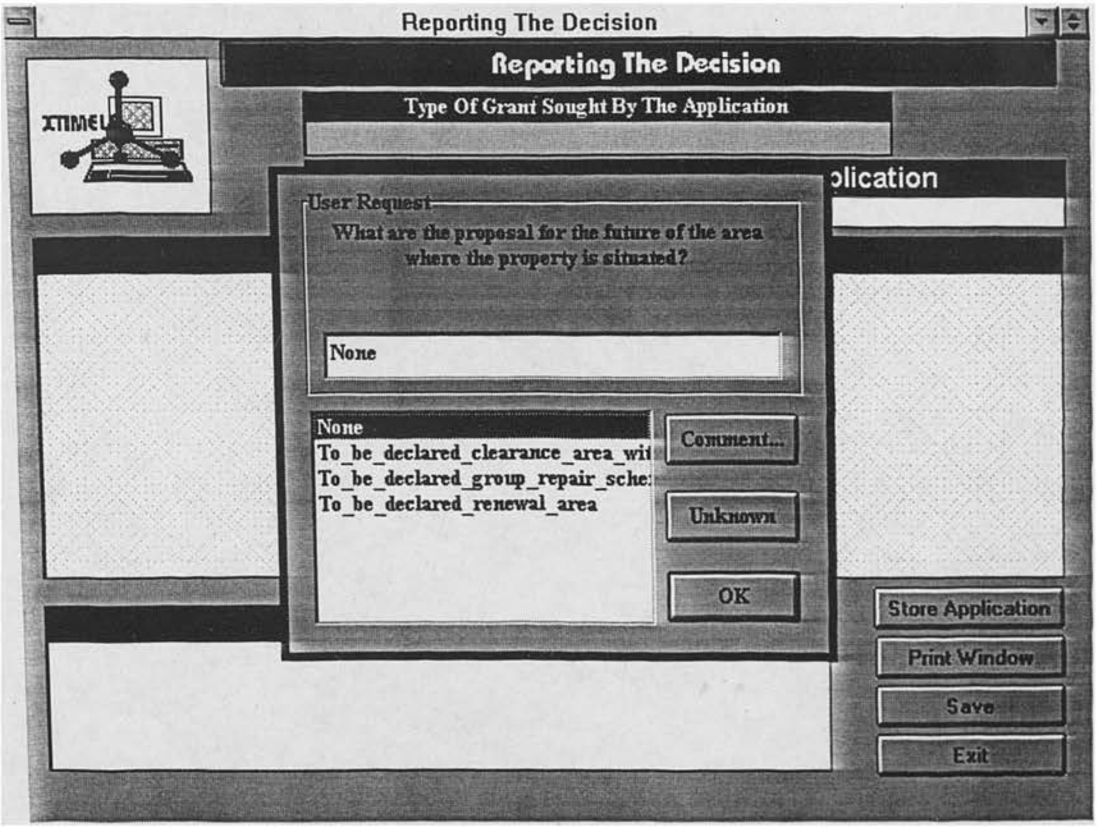


Screen 250: Input information for determing the degree of eligibility

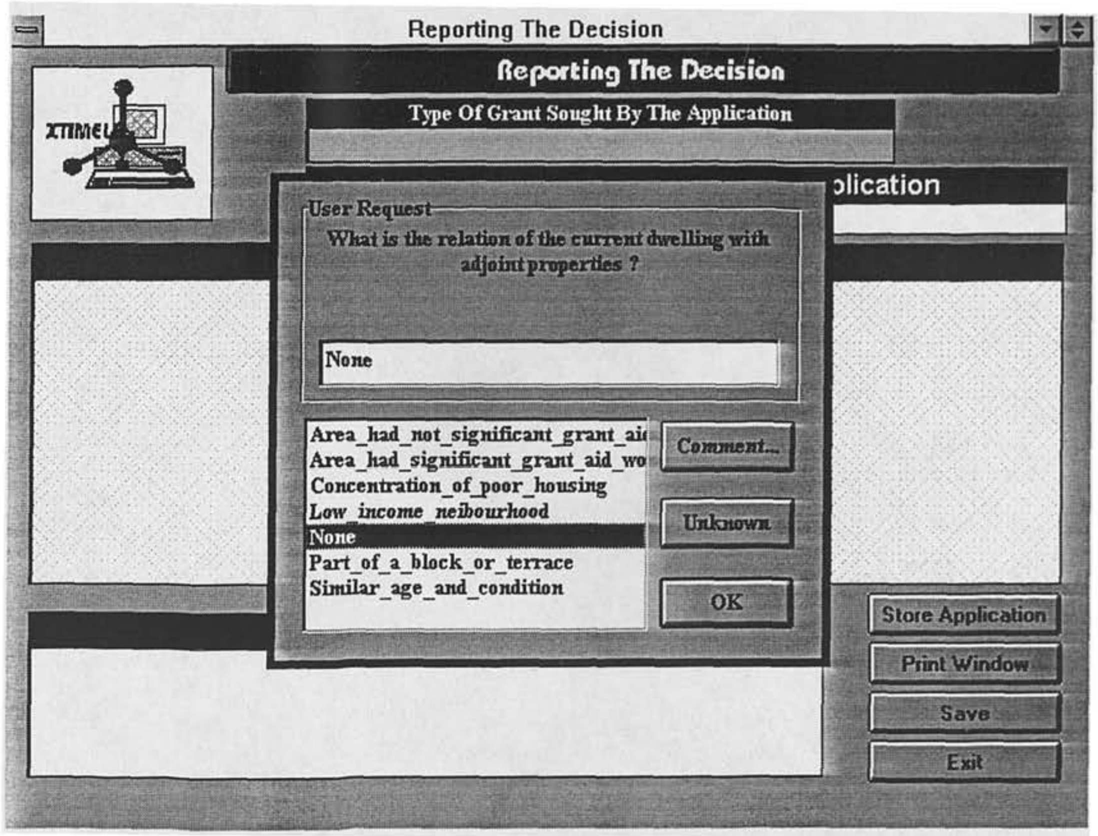


Screen 251: Report of the degree of eligibility of the application

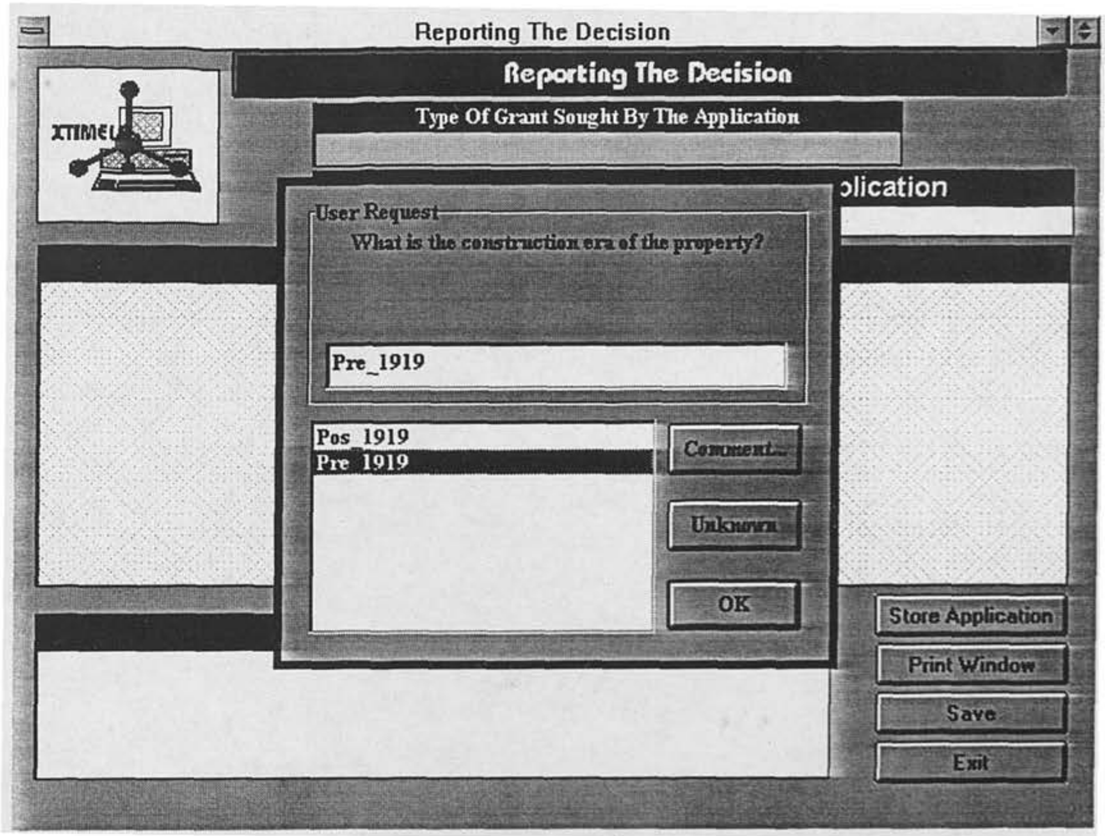
10.11- RETRIEVAL OF THE SOLUTION FOR THE CURRENT GRANT APPLICATION



Screen 252: Additional input information for retrieving a decision plan

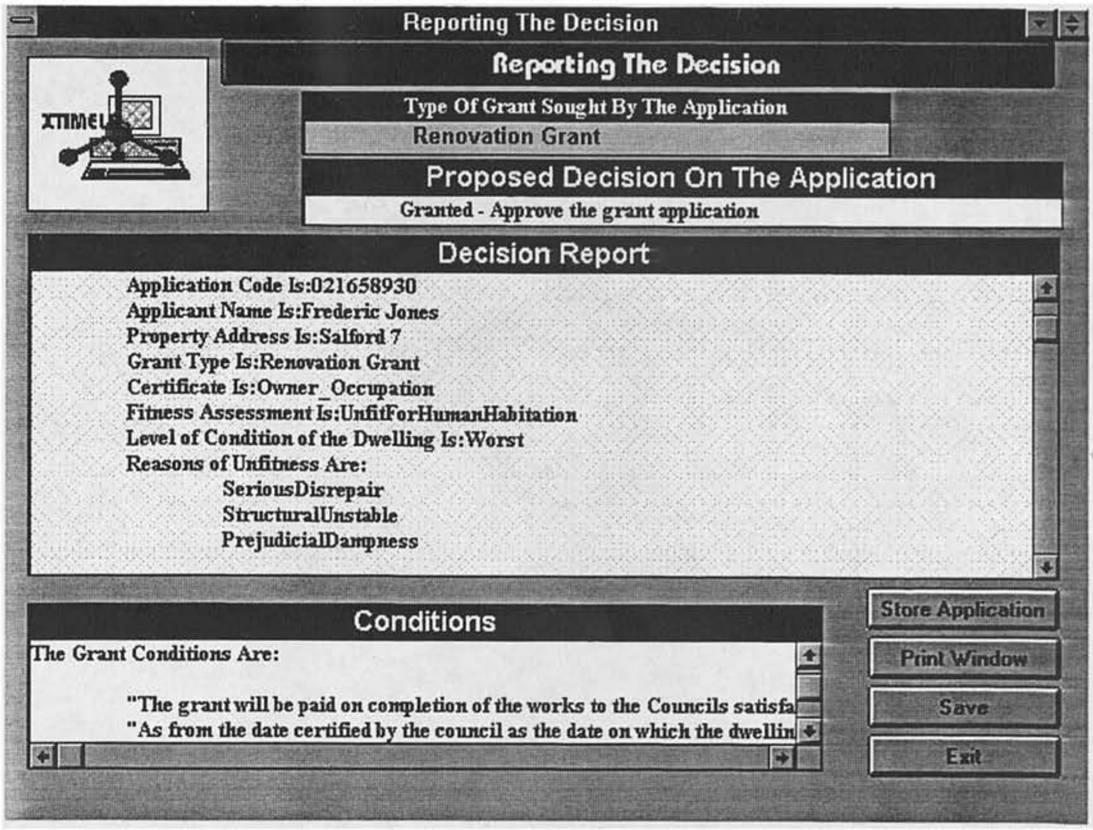


Screen 253: Additional input information for retrieving a decision plan



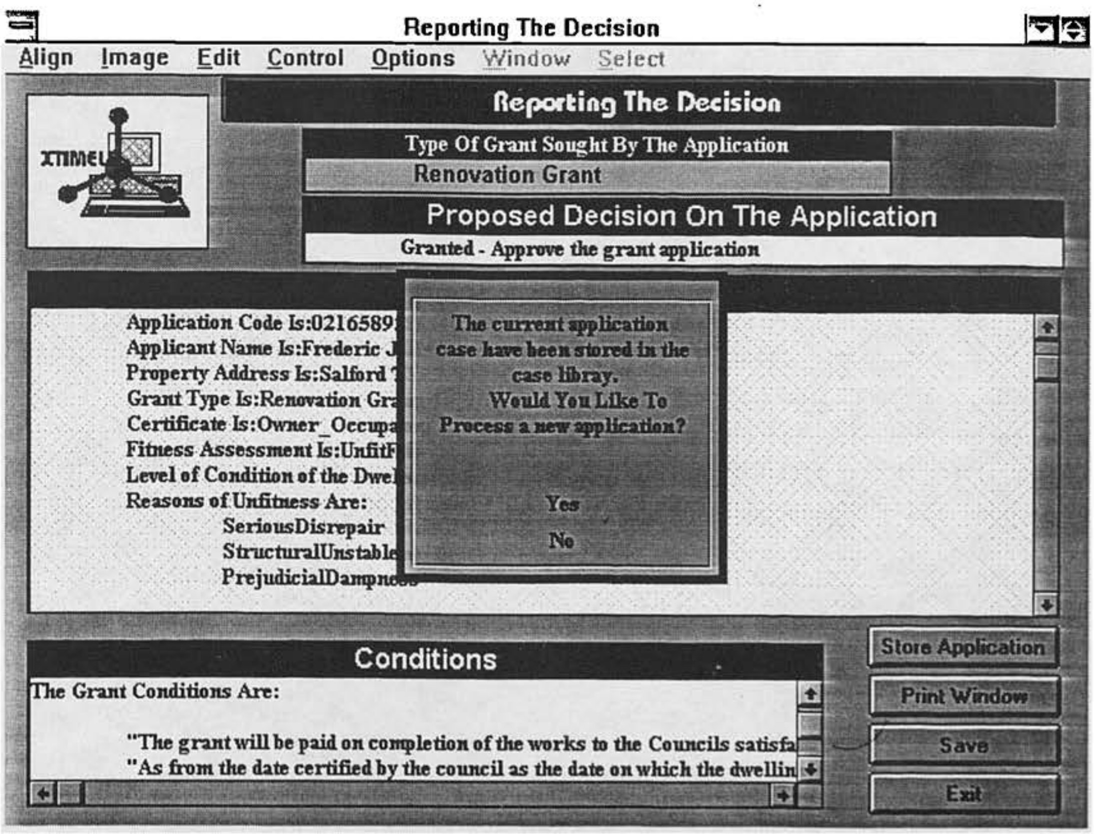
Screen 254: Additional input information for retrieving a decision plan

PRESENTATION OF THE SOLUTION PLAN



Screen 255: Presenting the solution plan for the current application

STORING THE CURRENT CASE IN THE CASE LIBRARY



Screen 256: Reporting that the current application has been stored

BIBLIOGRAPHICAL REFERENCES

Acorn T. and Walden S. (1992), SMART: Support Management Automated Reasoning Technology for Compaq Customer Service, in Innovative Applications of Artificial Intelligence, edited by A. Scott and Klahr P., pp. 3-18, Menlo Park, California, AAAI Press.

Act 1985, Housing Act 1985, Part IV, Chapter 69, pp. 2887-2970, HMSO Publication Centre, London 1985.

Act 1989, Local Government and Housing Act 1989, Part III, Chapter 42, pp. 2046-2386, HMSO Publication Centre, London 1990.

Addis T. R. (1993), Knowledge Science: A Pragmatic Approach to Research in Expert Systems, Cognitive Science & Human Computer Interaction 1993, pp. 371-389.

Allemang D. (1994), Combining Case-Based Reasoning and Task-Specific Architectures, IEEE Expert , Vol. 9 N°.5, October 1994, pp. 24-34.

Alterman R. (1986), Plexus in Case-Based Reasoning, edited by Kolodner J., pp. 609-610, Morgan Kaufmann Publishers, Inc. San Mateo California 1993.

Anderson J. (1993), Development of Expertise, in Readings in Knowledge Acquisition and Learning, pp. 61-77, Morgan Kaufmann Publishers, Inc. San Mateo California 1993.

Anick P. (1993), Integrating Natural Language Processing and Information Retrieval in a Troubleshooting Help Desk DEC, IEEE Expert, Vol. 8 N°. 8, December 1993, pp. 9-17.

Ashley K. and Rissland E. (1988), A Case-Based Approach To Modelling Legal Expertise, IEEE Expert, Vol. 3 N^{o}. 3, Fall 1988, pp. 70-77.

Ashley K. and Aleven V. (1992), Generating Dialectical Examples Automatically, in Proceedings of the Tenth National Conference on Artificial Intelligence, AAAI-92, July, pp. 654-660, AAAI Press/The MIT Press, San Jose, California 1992.

Bain W. (1989) JUDGE in Inside Case-Based Reasoning, edited by Riesbeck C. and Schank R., pp. 27-28, LEA Lawrence Erlbaum Associates, Publishers, Hillsdale, New Jersey, 1989.

Basden A. (1994), Three levels of benefits in expert systems, Expert Systems, Vol.11 N^{o}.2, May 1994, pp. 99-107.

Basden A., Watson I. and Brandon P. (1991), The Evolutionary Development Of Expert Systems, in Research and Development in Expert Systems VIII, in Proceedings of Expert Systems 91, the Eleventh Annual Technical Conference of the British Computer Society Specialist Group on Expert Systems, September 1991, pp. 67-81, London.

Beck H. (1991), Language Acquisition from Cases, in Proceedings: Case-Based Reasoning Workshop, May 1991 DARPA, pp. 159-169, Morgan Kaufmann Publishers, Inc. San Mateo, California.

Benjamins R. and Jansweijer W. (1994), Towards a Competence Theory of Diagnosis, IEEE Expert, Vol. 9 N^{o}.5, October 1994, pp. 24-34.

Birmingham W. and Klinker G. (1993), Knowledge-acquisition tools with explicit problem-solving models, The Knowledge Engineering Review, Vol. 8:1, pp. 5-25, 1993.

Brachman R. (1985), I Lied About the Trees, Or, Defaults and Definitions in Knowledge Representation, AI Magazine, Fall 1985, Vol.6 Nº.3, pp. 80-93.

Brandon P. (1993a), IT and Themes For The Nineties: Intelligence and Integration, in CIB W-65 Symposium 93, Organisation & Management of Construction The Way Forward, 7th International Symposium, Vol.1, September 1993, pp. 1-12, Port of Spain, Trinidad.

Brandon P. (1993b), Knowledge Based Systems Markets And Methodologies, in International Symposium On Economic Evaluation And The Built Environment 1993, CIB W55 and CIB W95, edited by Manso A., Newton S. and Gaarslev A. Vol. 6, pp. 11-27, LNEC, Lisbon 1993.

Brandon P., Basden A., Hamilton I. and Stockley J. (1988), Expert Systems The Strategic Planning Of Construction Projects, The RICS/ALVEY Research for Chartered Quantity Surveyors 1988, RICS Publications, London, UK 1988.

Branting L. and Porter B. (1991), Rules and Precedents as Complementary Warrants, in Proceedings of the Ninth National Conference on Artificial Intelligence, AAAI-91, July, Vol. 1, pp. 3-9, Anaheim, California 1991.

Brown D. and Chandrasekaran B. (1989), Design Problem Solving- Knowledge Structures and Control Strategies, Pitman Publishing, London 1989, ISSN 0268-7526.

Buchanan B. and Smith R. (1993), Fundamentals of Expert Systems in Readings In Knowledge Acquisition And Learning, edited by Buchanan and Wilkins, pp. 127-146, Morgan Kaufmann Publishers Inc. San Mateo California 1993.

Cervera E. (1993), KBS Validation: From Tools to Methodology, Guest Editor's Introduction, IEEE Expert, Vol. 8 Nº. 3, June 1993, pp. 45-47.

Chandrasekaran B. (1986a), Generic Tasks for knowledge-based reasoning. The right level of abstraction for knowledge acquisition, IEEE Expert, Vol. 1 N°. 3, Fall 1986, pp. 23-29.

Chandrasekaran B. (1986b), Generic Tasks in Knowledge-Based Reasoning: High-Level Building Blocks for Expert System Design, Readings In Knowledge Acquisition And Learning, edited by B. Buchanan & D. Wilkins, pp. 170-177, Morgan Kaufmann Publishers, Inc. 1993, San Mateo, California.

Chandrasekaran B. (1990), Design Problem Solving: A Task Analysis, AI Magazine, Vol. 11 No. 4, Winter 1990, pp. 59-71.

Chandrasekaran B., Johnson T. and Smith J. (1992), Task-Structure Analysis for Knowledge Modelling, Communications of the ACM, Vol. 39 N.° 9, September 1992, pp. 124-138.

Clancey W. (1985), Heuristic Classification, Artificial Intelligence Vol. 27 N°. 3 1985, pp. 289-350.

Clancey W. (1993), Acquiring, Representing, and Evaluating a Competence Model of Diagnostic Strategy, in Readings In Knowledge Acquisition And Learning, edited by B. Buchanan & D. Wilkins, pp. 178-215, Morgan Kaufmann Publishers, Inc. 1993, San Mateo, California.

Cragun J. and Steudel H. (1987), A Decision-based table processor for checking completeness and consistency in rule-based experts systems, International Journal of Management Studies (UK), 26 (5) May 1987, pp. 633-648.

Department of the Environment (1982-1992) Housing and Construction Statistics 1982-1992, March 1993, HMSO Books Publications Centre.

Department of the Environment (1990a), Local Government and Housing Act 1989, Area Renewal, Unfitness, Slum Clearance and Enforcement Action, Circular 6/90, HMSO Books Publications Centre.

Department of the Environment (1990b), Local Government and Housing Act 1989, House Renovation Grants, Circular 12/90 HMSO Books Publications Centre.

Department of the Environment (1991), English House Condition Survey: 1991, HMSO Books Publications Centre.

Department of the Environment (1993) Housing and Construction Statistics 1993, March 1994, HMSO Books Publications Centre.

Department of Trade and Industry, (1992), Innovation: Knowledge Based Systems; Survey of UK Applications, DTI, London 1992.

Domeshek E. (1991), What Abby Cares About, in Proceedings: Case-Based Reasoning Workshop, DARPA, May 1991, pp. 13-24, Morgan Kaufmann Publishers, Inc. San Mateo, California.

Feigenbaum E., Friedland P., Johnson B., Nii H., Schorr H., Shrobe H. and Engelmore R. (1994), Knowledge-Based Systems Research and Applications in Japan, 1992, AI Magazine, Vol. 15 No. 2, Summer 1994, pp. 29-43.

Feret M. and Glasgow J. (1993), Hybrid Case-Based Reasoning for the Diagnosis of Complex Devices, in Proceedings of the Eleventh National Conference on Artificial Intelligence, AAAI-93, Vol. 1, July 1993, pp. 168-175, Washington DC 1993.

Forsythe D. and Buchanan B. (1993) Knowledge Acquisition for Expert Systems: Some Pitfalls and Suggestions, in Readings In Knowledge Acquisition And Learning, edited by B. Buchanan & D. Wilkins, pp. 117-124, Morgan Kaufmann Publishers, Inc. 1993, San Mateo, California.

Giarratano J. and Riley G. (1994), Expert Systems, Principles and Programming, 2nd ed., PWS Pub. Co., 1994 Boston.

Goel A. (1989), Integration Of Case-Based Reasoning And Model-Based Reasoning For Adaptive Design Problem Solving, Dissertation for PhD award, Ohio Sate University 1989.

Goel A. and Chandrasekaran B. (1989), Functional Representation of Designs and Redesign Problem Solving, in Proceedings of the Eleventh International Conference on Artificial Intelligence, Vol. 2, pp. 1388-1394, Menlo Park, California.

Goel A., Kolodner J., Pearce M., Billington R. and Zimring C. (1991), Towards a Case-Based Tool for Adding Conceptual Design Problem Solving, Proceedings: Case-Based Reasoning Workshop, DARPA, May 1991, pp. 109-120, Morgan Kaufmann Publishers, Inc. San Mateo, California.

Goel A., Ali K., Donnellan M., Garza A. and Callantine T. (1994), Multistrategy Adaptive Path Planning, IEEE Expert , Vol. 9 N^{O}. 6, December 1994, pp. 57-65.

Goodman M. (1990), Prism in Case-Based Reasoning edited by Kolodner J., pp. 611-612 Morgan Kaufmann Publishers, Inc. San Mateo, California 1993.

Green C. and Keyes M. (1987), Verification And Validation Of Expert Systems, in Proceedings of the Western Conference on Expert Systems (WESTEX-87), IEEE Computer Society 1987, pp. 38-43.

Gupta U. (1991), Validating and Verifying Knowledge-Based Systems, The IEEE Computer Society Press 1991, pp. 176-187, Los Alamitos California.

Hammond K. (1986), CHEF: A model of case-based planning, in Proceedings of the National Conference on Artificial Intelligence, AAAI-86, August, Vol. 1, pp. 267-271, AAAI Press/MIT Press, Philadelphia, PA 1986.

Hammond K. (1989), Case-Based Planning: Viewing Planning as a Memory Task, Academic Press, Inc. Harcourt Brace Jovanovich, Publishers, New York 1989.

Hammond K. (1992), Reasoning as Remembering: The Theory and Practise of CBR, in Proceedings of the Tenth National Conference on Artificial Intelligence, AAAI-92, July, pp. 865, AAAI Press/ The MIT Press, San Jose, California 1992.

Hamscher W. (1991), Principles of Diagnosis: Current Trends and a Report on the First International Workshop, AI Magazine ,Vol. 12 N°. 4, Winter 1991, pp. 15-23.

Hennessy D. and Hinkle D. (1991), Initial Results from Clavier: A Case-Based Autoclave Loading Assistant, in Proceedings: Case-Based Reasoning Workshop, DARPA, May 1991, pp. 225-232, Morgan Kaufmann Publishers, Inc. San Mateo, California.

Hennessy D. and Hinkle D. (1992), Applying Case-Based Reasoning to Autoclave Loading, IEEE Expert, Vol. 7 N°. 5, October 1992, pp. 21-26.

Hinkle D. and Toomey C. (1995), Applying Case-Based Reasoning to Manufacturing, AI Magazine, Vol. 16 N°. 1, Spring 1995, pp. 65-73.

Hinrichs T. and Kolodner J. (1991), The Roles of Adaptation in Case-Based Design, in Proceedings: Case-Based Reasoning Workshop, DARPA May 1991, pp. 121-132 Morgan Kaufmann Publishers, Inc. San Mateo, California.

Hoppe T. and Meseguer P. (1993), VVT Terminology: A Proposal, IEEE Expert, Vol. 8 N°.3, June 1993, pp. 48-55.

IntelliCorp (1992), Kappa-PC User Guide, Early Release Draft IntelliCorp, Inc. April 1992, USA 1992.

Irgon A., Zolnowski J., Murray K. and Bellcore M. (1990), Expert System Development, IEEE Expert, Vol. 5 N°.3, June 1990, pp. 25-40.

Jafar M. and Bahill A. (1993), Interactive Verification of Knowledge-Based Systems, IEEE Expert, Vol. 8 N°.1, February 1993, pp. 25-32.

Kitano H., Shimazu H. and Shibata A. (1993), Case-Method: A Methodology for Building Large-Scale Case-Based Systems, in Proceedings of the Eleventh National Conference on Artificial Intelligence, AAAI-93, July, pp. 303-308, Washington DC 1993.

Klein G. and Brezovic C. (1988), Evaluation of expert systems, in Defence Applications of Artificial Intelligence, edited by S. Andriole and G. Hopple, pp. 111-137, Lexington Books, Heath, Washington DC, 1988.

Klinker G., Bentolila J., Genetet, Grimes M. and McDermott J. (1987), KNACK- Report Driven Knowledge Acquisition, International Journal of Man-Machine Studies Vol. 26 (1), pp. 65-79, 1987.

Kolodner J. (1991), Improving Human Decision Making through Case-Based Decision Aiding, AI Magazine, Vol. 12 N°. 2, Summer 1991, pp. 52-68.

Kolodner J. (1993), Case-Based Reasoning, Morgan Kaufmann Publishers, Inc., 1993, ISBN1-55860-237-2.

Kolodner J. and Riesbeck K. (1986), Experience, Memory, And Reasoning, Lawrence Erlbaum Associates, Publishers, 1986.

Kolodner J. and Mark W. (1992), Case-Based Reasoning, Guest Editors' Introduction, IEEE Expert, Vol. 7 N°.5, October 1992, pp. 5-6.

Koton P., (1988), Reasoning about evidence in causal explanation, in Proceedings of the National Conference on Artificial Intelligence, AAAI-88, Vol. 1, pp. 256-261, AAAI /Press MIT Press, St. Paul, MN 1988.

Kriegsman M. and Barletta R. (1993), Building a Case-Based Help Desk Application, IEEE Expert, Vol. 8 N°. 3, December 1993, pp. 18-26.

Leake D. (1991), ACCEPTER A Program for dynamic similarity assessment in case-based explanation, in Proceedings: Case-Based Reasoning Workshop, DARPA May 1991, pp. 51-62 Morgan Kaufmann Publishers, Inc. San Mateo, California.

Leake D. (1994), AAAI-93 Workshops: Summary report, AI Magazine, Vol. 15 N°.1, Spring 1994, pp. 63-64.

Maher M. (1990), Process Models for Design Synthesis, AI Magazine, Vol. 11 N°. 4, Winter 1990, pp. 49-58.

Kolodner J. (1991), Improving Human Decision Making through Case-Based Decision Aiding, AI Magazine, Vol. 12 N°. 2, Summer 1991, pp. 52-68.

Kolodner J. (1993), Case-Based Reasoning, Morgan Kaufmann Publishers, Inc., 1993, ISBN1-55860-237-2.

Kolodner J. and Riesbeck K. (1986), Experience, Memory, And Reasoning, Lawrence Erlbaum Associates, Publishers, 1986.

Kolodner J. and Mark W. (1992), Case-Based Reasoning, Guest Editors' Introduction, IEEE Expert, Vol. 7 N°.5, October 1992, pp. 5-6.

Koton P., (1988), Reasoning about evidence in causal explanation, in Proceedings of the National Conference on Artificial Intelligence, AAAI-88, Vol. 1, pp. 256-261, AAAI /Press MIT Press, St. Paul, MN 1988.

Kriegsman M. and Barletta R. (1993), Building a Case-Based Help Desk Application, IEEE Expert, Vol. 8 N°. 3, December 1993, pp. 18-26.

Leake D. (1991), ACCEPTER A Program for dynamic similarity assessment in case-based explanation, in Proceedings: Case-Based Reasoning Workshop, DARPA May 1991, pp. 51-62 Morgan Kaufmann Publishers, Inc. San Mateo, California.

Leake D. (1994), AAAI-93 Workshops: Summary report, AI Magazine, Vol. 15 N°.1, Spring 1994, pp. 63-64.

Maher M. (1990), Process Models for Design Synthesis, AI Magazine, Vol. 11 N°. 4, Winter 1990, pp. 49-58.

Maher M. and Balachandran B. (1994), Structural Design, Journal of Computing in Civil Engineering, ASCE, Vol. 8 N.º 3, July 1994, pp. 359-375.

Marcot B. (1987), Testing Your Knowledge Base, AI Expert, Vol. 2 Nº. 8, August 1987, pp. 42-47, Miller Freeman Publications.

McDermott J. (1988), Preliminary steps towards a taxonomy of problem solving methods, Automating Knowledge Acquisition for Expert Systems, pp. 225-256, Boston 1988.

McDermott J. (1993), Preliminary Steps Toward a Taxonomy of Problem-Solving Methods, in Readings in Knowledge Acquisition and Learning, edited by B. Buchanan & D. Wilkins, pp. 149-168, Morgan Kaufmann Publishers, Inc., San Mateo, California 1993.

McGraw K. and Harbison-Briggs K. (1989), Knowledge Acquisition Principles and Guidelines, New Jersey, Prentice Hall 1989.

Meseguer P. (1993), Expert System Validation Through Knowledge Base Refinement, 13th International Joint Conference on Artificial Intelligence IJCAI 93, Aug./Sep., Vol. 1, pp. 477-482 Chambery, France 1993.

Miyashita K. and Sycara K. (1993), A Case-based incremental schedule revision, in Knowledge-Based Scheduling, edited by M. Fox and M. Zweben, Morgan Kaufmann, 1993.

Motoda H., (1994) Second-Generation Expert Systems, IEEE Expert, Vol. 9 Nº.2, April 1994, pp. 66-75.

Moulin B. and Rousseau D. (1992), Automated Knowledge Acquisition from Regulatory Texts, IEEE Expert, Vol. 7 Nº. 5, October 1992, pp. 27-35.

Nasser J. (1988), Nuclear Power Plant Expert System Verification And Validation in Proceedings of the Workshop on Validation and Verification of Expert Systems, IEEE August 1988, pp. 26-41.

Navinchandra D. (1992), Innovative design systems, where are we and where do we go from here? Part I: Design by association, in The Knowledge Engineering Review, Vol. 7, pp. 183-213, 1992.

Newell A. (1982), The Knowledge Level, Artificial Intelligence, 18 (1), pp. 87-127.

Newell A., Yost G., Laird J., Rosenbloom P. and Altmann E. (1991), Formulating the Problem Space Computational Model, in Carnegie Mellon Computer Science: A 25- Year Commemorative, Rashid E. ed., ACM Press: Addison-Wesley, pp. 255-293, Reading Mass. Pa. 1991.

Newell A. and Chandrasekaran B. (1993), Expert Interview, IEEE Expert, Vol. 8 N°.3, June 1993, pp. 5-12.

Nguyen T., Perkins W., Laffey T. and Pecora D. (1987), Knowledge-base Verification, AI Magazine, Vol. 8 No. 2, Summer 1987, pp. 69-75.

Nguyen T., Czerwinski M. and Lee D. (1993), Compaq QUICKSOURCE Providing the Consumer with the Power of AI, AI Magazine, Fall 1993, Vol. 14 N°.3, pp. 50-60.

O'Hara K. and Shadbolt N. (1993), AI Models as a Variety of Psychological Explanation, 13th International Joint Conference on Artificial Intelligence IJCAI 93, Vol.1, Aug./Sep., pp. 188-193, Chambery, France 1993.

O'Keefe R., Balci O. and Smith E. (1987), Validating Expert System Performance, IEEE Expert, Winter 1987, Vol. 2 N°.4, pp. 81-90.

Oxman R.E. (1991), Case-Based Reasoning in Knowledge-Based Design, in Proceedings of the European Symposium on Management, Quality and Economics in Housing and other Building Sectors, pp. 1516-1525, Lisbon 1991.

Pearce M., Goel A., Kolodner J., Zimring C., Sentosa L. and Billington R. (1992), Case-Based Design Support, IEEE Expert, Vol. 7, N°.5, October 1992, pp. 14-20.

Preece A. (1990), Towards a methodology for evaluating expert systems, Expert Systems, Vol. 7, N°. 4, November 1990, pp. 215-236.

Puerta A., Tu S. and Musen M. (1992), Modelling Tasks with Mechanisms Technical Report KSL- Report 92-30, Knowledge Systems Laboratory, Stanford University, 1992.

Regoczei S. and Plantinga E. (1988), Creating the Domain of Discourse: Ontology and Inventory, Knowledge Acquisition Tools for Knowledge-Based Systems, edited by B. Gaines and J. Boose, Vol. 2, pp. 293-308, Academic Press 1988.

Reisbeck C. and Schank R. (1989), Inside Case-Based Reasoning, Lawrence Erlbaum Associates, Inc., Hillsdale, New Jersey, NJ, 1989.

Rissland E., Basu C., Daniels J., McCarthy J., Rubinstein Z. and Skalak D. (1991), A Blackboard-Based Architecture for CBR: An Initial Report, in Proceedings: Case-Based Reasoning Workshop, DARPA, May 1991, pp. 77-92, Morgan Kaufmann Publishers, Inc. San Mateo, California.

Rissland E., Skalak D. and Friedman M. (1993), Case Retrieval trough Multiple Indexing and Heuristic Search, in Proceedings of the 13th International Joint Conference On Artificial Intelligence IJCAI 93, Aug./Sep., Vol. 2, pp. 902-908, Chambery, France 1993.

Sanders K. (1991), Within the Letter of the Law: reasoning among multiple cases, in Proceedings: Case-Based Reasoning Workshop, DARPA, May 1991, pp. 317-326, Morgan Kaufmann Publishers, Inc. San Mateo, California.

Schreiber G., Wielinga B. and Breuker J. (1993), KADS: A Principled Approach to KBS Development, Academic Press Inc. San Diego, 1993, ISBN 0-12-629040-7.

Sharma R. S. (1992), Evaluating expert systems: the socio-technical dimension of quality, Expert Systems, Vol. 9 N^{o}. 3, August 1992, pp. 125-137.

Shaw M. and Woodward J. (1990) Modelling expert knowledge, in Readings in Knowledge Acquisition and Learning 1993, edited by Buchanan G. & Wilkins D., pp. 78-91, Morgan Kaufmann Publishers, Inc., San Mateo California.

Simoudis E. (1992), Using Case-Based Retrieval for Customer Technical Support, IEEE Expert, Vol. 7 N^{o}.5, October 1992, pp. 7-12.

Simoudis E. and Miller J. (1991), The Application of CBR to Help Desk Applications in Proceedings: Case-Based Reasoning Workshop, DARPA, May 1991, pp. 25-36, Morgan Kaufmann Publishers, Inc. San Mateo, California.

Skousen R. (1989), Analogical Modelling of Language, Kluwer Academic Publishers, Boston MA, 1989.

Slade S. (1991), Case-Based Reasoning: A Research Paradigm, AI Magazine, Vol. 12 N°. 1, Spring 1991, pp. 42-55.

Small R. and Yoshimoto B. (1995), The VLS Tech-Assist Expert System, AI Magazine, Vol. 16 N°.1, Spring 1995, pp. 41-50.

Smith J. and Johnson T. (1993), A Stratified Approach to Specifying, Designing, and Building Knowledge Systems, IEEE Expert, Vol. 8, N°.3, June 1993, pp. 15-25.

Steels L. (1990), Components Of Expertise, AI Magazine, Summer 1990, Vol. 11 N°. 2, , pp. 29-49.

Strosnider J. and Paul C. (1994), A structured View of Real-Time Problem Solving, AI Magazine, Vol. 15 N°. 2 , Summer 1994, pp. 45-66.

Suwa M., Scott A., and Shortliffe E. (1982), An approach to verifying completeness and consistency in a rule-based expert system, AI Magazine, Fall 1982, Vol. 3 N°. 4, , pp. 16-21.

Sycara K. (1987), PERSUADER in Influences: A Thematic Abstraction for Creative Use of Multiple Cases, in Proceedings: Case-Based Reasoning Workshop, DARPA, May 1991, pp. 133-144 , Morgan Kaufmann Publishers, Inc. San Mateo, California.

Tuthil S. (1990), Knowledge Engineering- Concepts and Practices for Knowledge-Based Systems, TAB professional Reference Books, Division TAB Books Inc., Blue Ridge Summit, PA.

Vargas J. and Raj S. (1993), Developing maintainable experts systems using case-based reasoning, Expert Systems, Vol. 10 N°. 4, November 1993, pp. 219-226.

Veloso M. and Carbonell J. (1991) Variable-Precision Case Retrieval in Analogical Problem-Solving, in Proceeding: Case-Based Reasoning Workshop, DARPA, May 1991, pp. 93-106, Morgan Kaufmann Publishers, Inc. San Mateo, California.

Watson I., Basden A. and Brandon P. (1992a), The Client-Centred Approach: Expert Systems Development, Expert Systems, The International Journal Of Knowledge Engineering, Vol. 9 N°. 4, November 1992, pp. 181-188.

Watson I., Basden A. and Brandon P. (1992b), The Client-Centred Approach: Expert System Maintenance, Expert Systems, The International Journal Of Knowledge Engineering, Vol. 9 N°. 4, November 1992, pp. 189-196.

Watson I. and Brandon P. (1993), Strategic Maintenance Prediction: An Expert System For Facilities Management, in CIB W-65 Symposium 93, Organisation & Management of Construction The Way Forward, 7th International Symposium, Vol. 3, September 1993, pp. 1087-1095, Port of Spain, Trinidad.

Wielinga B., Schreiber and Breuker J. (1993), KADS: a modelling approach to knowledge engineering, in Readings In Knowledge Acquisition And Learning, edited by B. Buchanan & D. Wilkins, pp. 92-119, Morgan Kaufmann Publishers, Inc. 1993, San Mateo, California.

Yen J. and Lee J. (1993) A Task-Based Methodology for Specifying Expert Systems, IEEE Expert, Vol. 8 N°.1, February 1993, pp. 8-15.

Yost G. (1993), Acquiring Knowledge in Soar, The Soar architecture, IEEE Expert, Vol. 8 N°. 3, June 1993, pp. 26-34.

Zhao F. and Maher M., (1988), Using analogical reasoning to design buildings, in Engineering with Computers, Vol. 4, pp. 107-119.

Zito-Wolf R. and Alterman R., (1993), A Framework and an Analysis of Current Proposals for the Case-Based Organisation and Representation of Procedural Knowledge, in Proceedings of the Eleventh National Conference on Artificial Intelligence, AAAI-93, July, pp. 73-78, AAAI Press/The MIT Press, Washington, DC 1993.

Zozaya-Gorostiza C., Hendrickson C. and Rehak D. (1989), Knowledge-Based Process Planning for Construction and Manufacturing, Department of Civil Engineering Design Research Center Carnegie Mellon University, Pittsburgh, Pennsylvania, Academic Press Inc., 1989.