

# **JAD-CASE.**

## **Tasks and Documents Specification**

### **for Requirements Engineering Process**

*Balda, María Verónica*<sup>1</sup>

*Vicenzi, Ana Laura*<sup>2</sup>

*Bertone, Rodolfo*<sup>3</sup>

*De Giusti, Armando*<sup>4</sup>

*L.I.D.I.*<sup>5</sup>

*Laboratorio de Investigación y Desarrollo en Informática*

*Depto. Informática*

*Fac. Ciencias Exactas*

*Universidad Nacional de La Plata*

### **Abstract**

The structured techniques for systems analysis and design are still the most widely known. One of the alternatives of this technique is the JAD methodology (Joint Application Design). This methodology is an improvement with respect to the existing ones - such as classical life cycle, spiral development, increasing prototypes - since it expands the role of the users, and allows them to define their own objectives, requirements and external design.

Problem definition, as well as requirements initial analysis, are the basis for the whole software engineering process; the success of the developed system depending mostly on these two initial stages. This is the reason why the use of a methodology emphasizing these initial stages is important, since it would provide a higher quality design, easy to maintain and modify.

The proposed objective was the construction of a CASE tool which would include the JAD methodology within an environment where participants (users and analysts) can interact to manage the flow of tasks and documents generated in the Requirements Engineering stage.

---

<sup>1</sup> Chief Assistant, part-time, Researcher (e - mail: [mbalda@info.unlp.edu.ar](mailto:mbalda@info.unlp.edu.ar))

<sup>2</sup> Graduate Assistant, part-time, Researcher (e - mail: [avicenzi@info.unlp.edu.ar](mailto:avicenzi@info.unlp.edu.ar))

<sup>3</sup> Assistant Professor, full-time, Researcher (e - mail: [pbertone@info.unlp.edu.ar](mailto:pbertone@info.unlp.edu.ar))

<sup>4</sup> Chair Professor, full-time. Principal Researcher, CONICET. (e-mail: [degiusti@info.unlp.edu.ar](mailto:degiusti@info.unlp.edu.ar))

<sup>5</sup> 50 y 115 1º Piso, (1900) La Plata, Buenos Aires, Argentina  
TE-FAX: 54-21-22-7707

## Introduction

The costs associated with a software project can be divided into three main stages: [BIA97]

- research: this stage involves the initial analysis of requirements, feasibility studies, planning and detailed analysis.
- engineering: software production is involved, from a high-level design up to codification, including final-systems proofing.
- maintenance: users support is included here, as well as the efforts needed for system modification (corrective, perfective, adaptive).

The effects caused by mistakes or omissions during the initial research stage hardly affect the final results if detected during this same stage. Now, this cost considerably increases if the problems are detected during the engineering stage; and their growth becomes exponential if they pass undetected to the final, maintenance stage.

A good methodology during the engineering stage is vital for the software project to be successful, but it needs to be based on a correct and complete Analysis stage. Furthermore, software projects cost reduction is still based, and on an increasing basis, on the research stage.

One of the main problems with software development methodologies is, still, the role of the final users. Isolated surveys are carried out in order to obtain the requirements. Thus, a technical team evaluates the basic needs of the users, and then uses the results in order to develop the final product. [PRE93].

The participation of users during this first, highly relevant stage is not promoted enough. An increasing user participation should be sought for from both sides: users should increase their participation and analysts should take users more into account. Some of the problems are due to the fact that users *feel that they are wasting their time* when the analyst asks too many questions; and analysts feel that they have doubts about how to solve certain situations presented by the users. The result is a non-exact and generally incomplete requirements analysis, which in time ends up in a system which does not comply with users expectations. [YOU93].

This loss of time and effort can be avoided if the gap between user and analyst is eliminated, or at least minimized. There are a series of ideas to do this [BAL97]:

- to gather, structure and model the requirements with a strictly defined methodology.
- to properly study and register the information, in order to define the *what* of the problem.
- to incorporate the tools needed in order to facilitate the requirements analysis activity.

If, in addition to this, we consider that a changes component will invariably be present, the development and management of a software product requirements turns out to be a task with a high management and information-handling content, needed for a *correct* development of a system. [OLS91].

The solution proposed is deeply related to two concepts:

- Joint Application Design (JAD)
- Workflow

### Joint Application Design (JAD)

JAD is a methodology which allows to obtain the external design of a software product from the specification of the system general objectives. It encourages team-work among users and systems personnel, who, by means of previously organized meetings, outline and understand the basic and specific needs of the software to be developed. Thus, both groups are engaged in the development of the system and share the responsibility of the eventual drawbacks that the final system might show.

The advantages of JAD when compared with the traditional methodologies is related with the application of the following basic principles [AUG91]:

- *Group dynamics*: under the guidance of an experienced group leader, the selected work team analyzes requirements, generates ideas and makes decisions leading to the obtention of the system external design (quick prototypes).
- *Rational and organized process*: the top-down analysis method is used, and tasks are divided into stages in order to carry out the production of objectives, requirements and external design. The top-down analysis reduces design gaps, and the designer is able to consider each stage individually and thus improve them as necessary. As a consequence, the specific tasks and outputs are done in the appropriate sequence, resulting in an organized and productive process.
- *WYSIWYG documentation method*: the use of methodology throws as a result documents including all the ideas and decisions generated during the group sessions. This is why participants are familiar with their format and contents, which facilitates contents revision and approval, as well as technical design, codification, and testing tasks performed by the information systems team.
- *Visual aids*: among the numerous visual-aid devices included in the methodology are prototypes, graphics, transparencies, boards, and special magnetic boards, which are used to communicate and validate ideas in a better way during the design process.

JAD methodology defines six different categories of participants:

- *Session leader*: the person in charge of guiding participants through the process. He organizes the JAD, facilitates the group dynamics, and shares his experience in the use of methodology. He is in charge of tasks planning and follow-up.
- *Analyst*: his main responsibility is the production of documents. In addition to this, he contributes with his own ideas to the session discussions, assists and supports the session leader, and develops the prototype.
- *Project manager*: he has full authority over the project. He makes decisions according to the organization limits and objectives and checks and approves JAD documents.
- *User*: together with other participants, he examines the organization information-processing needs and generates new alternatives for the outlined requirements. He must be experienced and familiar with the organizational environment.
- *Information systems personnel*: they give advice to the other participants as to how to take advantage of the information systems resources, and guide them in order to avoid the development of an ideal, non-implementable design.

- *Specialist*: he provides his experience on a specific topic.

The JAD assures a team work among participants, who define the project requirements and scope before dealing with system details. Their activities are divided into two fundamental stages: *plan* and *design*.

The plan stage initiates the project and it includes objectives definition. During its development the following activities must be performed:

- *Identification of the system high-level requirements.*
- *Definition and boundaries of the scope of the system.*
- *Planning of the design stage.*
- *Publication and approval obtention of plan documents.*

The design stage includes detailed requirements, user interface, and relationship with other systems, which are based on the document obtained during the plan stage. The objectives of this stage are:

- *Detailed definition of the requirements and scope of the system.*
- *Design of screen and reports layouts.*
- *Capture of edition, validation, processing, and interface requirements.*
- *Development of a prototype.*
- *Completion and approval obtention of the design stage document.*

For both plan and design stages, the tasks are grouped together in three logically related phases: customization, session, and wrap-up. A project developed following the JAD methodology will consist of a repetitive series of these three phases. If the project includes more than one design activity, a final wrap-up phase is added, where a final, general revision of the design is done.

The *customization* phase is the key for an effective software design JAD. It includes customization tasks for the following session phase, mostly defined by the session leader and one or two analysts.

During the *session* phase of the plan stage, the system objectives and requirements are recognized; and during the design stage, the external design is created, representing the way users see the system. This session method is the one that makes the JAD methodology different from any other one, since it uses an effective way to collect requirements and to design the system. It includes both experienced users and information systems experts in its design team, which allows the discussion of their ideas in order to obtain improvements leading to a high-quality design.

The *wrap-up* phase produces JAD's formal outputs, which means that a document showing the decisions generated during the session is obtained. This document is checked by the participants of the session. Once updated, the final JAD plan document is handed out to the design participants and to the project manager. The final JAD design document will be handed out to and used by the technical team to develop the system inner design. A prototype is another output produced during the JAD design wrap-up phase, and it includes every screen template designed during the session phase. The final output is a presentation to the project manager, where the results of the session are explained.

## Workflow Technology

Currently, business companies must handle a global competitiveness, reduce business costs, and develop new services and products quickly. In order to be able to cope with these requirements, they must constantly reconsider and optimize the way in which they do business, and change their applications and information systems so that they support evolving business processes. Workflow is one of the technologies which is able to handle these requirements by means of the provision of methodologies and software providing the following: [GEO94]

- modeling of business processes to capture them as workflow specifications.
- re-engineering of business processes to optimize the specified processes.
- automation of the workflow to generate implementations of the workflow from specifications.

A workflow is a system where the tasks are performed either in a serial way or in parallel by two or more members of a work group to achieve a common objective. The idea of the workflow is its application to the automation of routine tasks in order to devote the extra free time available to other more *creative* tasks. [ALO95] [WEB] [WAL95] [BRI93]

Workflow tools help with the modeling of new processes and provide a suitable environment for their simulation, thus allowing to compare existing platforms, processes, and applications. This software provides an excellent environment for documents handling, thus facilitating version control, and keeping the security levels [MAY94] [JOO95]. Its characteristics are:

- concentration on the interaction between the actors (Individuals, groups and/or machines) rather than on activities, which shows that it is a combination of *computer sciences* and *business sciences*.
- different types of people and units from the organization are involved.
- limits may be inside or outside the organization.
- procedures and tools integration.

One of the most important characteristics is integration. There are *Legacy Systems* in any organization which must be incorporated to the workflow due to the need of using *elements* from one application in another one. These *legacy systems* can be triggered from the workflow any time the information they access or produce might be needed. [RAM96] [BER96]

Within an office, there are several products (Word Processors, Data Bases, Templates, E-Mail, design Tools, protocols, etc.) which must work in a joint way in order to allow that a process created with one product can be exchanged and used for the definition of other processes using other products. [EDG95].

When this tool is used for modeling, there is some information which should be available: which data are necessary (location and owner), when they are needed (plans, delays, origin), who needs them (users, roles, etc.) and how to control their end.

The basic functions which should provide the different tools of a workflow environment are:

- methods to answer other users requirements.
- "beginning of task" messages.

- easy access to the information generated in the circuit.
- help for the decision making process in order to make it easy, by means of access to data bases, and different documents and messages.
- easy-to-create documents.
- distribution of the documents generated.
- friendly interfaces.

## Objectives

The general objective proposed was the definition and implementation of an environment to manage the tasks and documents flow involved in the requirements process. These tasks can be summarized as follows:

- management of system requirements evaluation meeting agendas.
- management of the generated documentation.
- management of communications among project members.
- management of the roles of the people involved in the project within each of the process tasks.
- creation of control points for each of the tasks
- management of documents versions according to changes.

## Solution specification

Once the JAD main functions were analyzed and understood, our goal was the automation of those tasks by managing its flow and the documents it generates. This lead to the obtention of an experimental CASE tool to facilitate the process. The environment manages and validates the resulting information of each task, and controls their execution in the established time and sequence. To achieve the objective the tool allows the following:

- definition of the project participants, and interaction with them. They will be selected among the members of the organization, and each of them will be assigned a role within the project. In addition to this, external<sup>6</sup> users may be added.
- standard or user-defined document formats for each particular application. The tool initially considers the use of a standardized documentation according to policies followed by most authors on Software Engineering. In addition to this, the tool allows to define within each project the format of the documents to be used.
- adjustment of the tasks circuit. A base circuit is provided, which may be adjusted according to the characteristics of each project. This adjustment involves the removal of some pre-defined steps which may be non-significant. It can also involve a change in the duration of each task.
- management of the tasks execution circuits, controlling their execution deadline.
- management of the communications among the participants (mail).
- management of documentation versions.

---

<sup>6</sup> External user: the person requesting the construction of a specific software.

## Solution Implementation

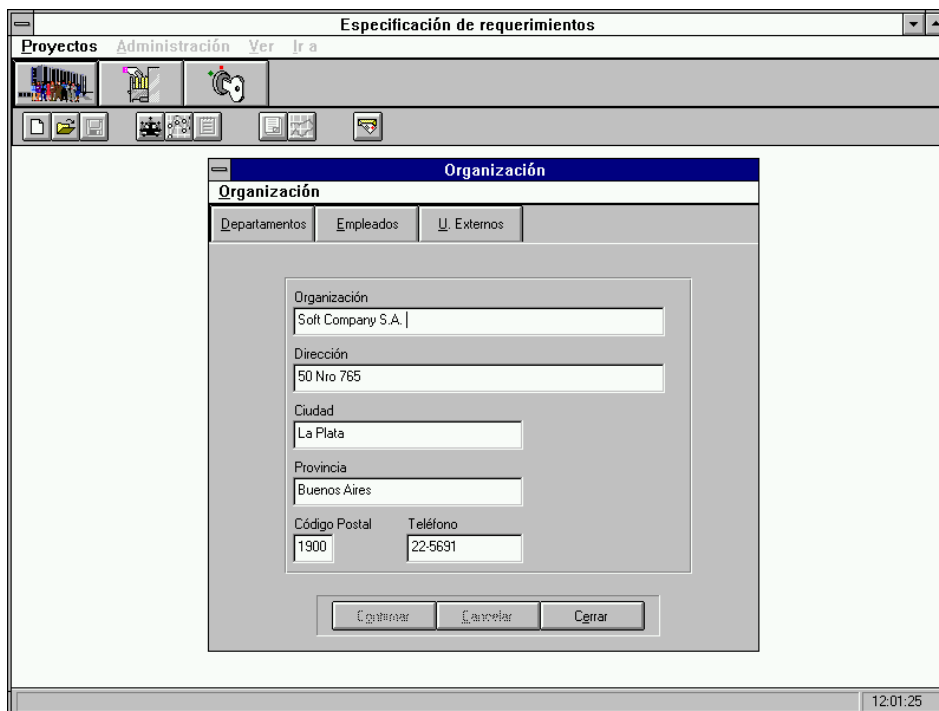
The implemented solution is divided into two large modules, which allow the JAD-CASE general management. These modules are:

- TOOL MANAGEMENT
- PROJECTS MANAGEMENT

### Tool management

The system basic definitions are in charge of a Manager, who must enter the company general information to the CASE. The activities performed by the manager are, in general, similar to those performed by a network manager. They can be summarized as follows:

- definition of the company general data (employees, departments, relationship between these two groups, external users, etc.) (figure 1)
- definition of the CASE tool users<sup>7</sup> (passwords, access<sup>8</sup> levels) (figure 2)



The screenshot shows a graphical user interface window titled "Especificación de requerimientos". The window has a menu bar with "Proyectos", "Administración", "Ver", and "Ir a". Below the menu bar is a toolbar with several icons. The main content area displays a form titled "Organización". The form has three tabs: "Departamentos", "Empleados", and "U. Externos". The "Empleados" tab is selected. The form contains several text input fields: "Organización" (containing "Soft Company S.A."), "Dirección" (containing "50 Nro 765"), "Ciudad" (containing "La Plata"), "Provincia" (containing "Buenos Aires"), "Código Postal" (containing "1900"), and "Teléfono" (containing "22-5691"). At the bottom of the form are three buttons: "Continuar", "Cancelar", and "Cerrar". The system clock in the bottom right corner shows "12:01:25".

Figure 1

### Projects management

The tool is developed to allow a simultaneous management of several projects. Each of them will have a manager and/or session leader. In order to manage the JAD-CASE project, it provides the following elements:

- DOCUMENTS SCHEMES MANAGER
- PARTICIPANTS MANAGER

<sup>7</sup> "Tool users" involves the company employees and external users.

<sup>8</sup> There are two access levels: complete (which authorizes the user to create new projects), and restricted (authorizes the manipulation of existing projects).

- WORKFLOW MANAGER
- WORKFLOW

### ***Documents Scheme Manager***

JAD-CASE provides standard documentation to be used at each stage of the project. This documentation was obtained from the main analysis and design characteristics obtained from several authors. Now, certain projects may require the management of specific information, which will not be necessarily included in the standard documents. Therefore, a documents schemes manager is required in order to carry out the necessary changes on the standardized documentation.

The manager allows to add, modify or remove elements to and from documents. This is done by changing the pre-defined templates of the used text editor.

Only the project managers or session leaders are authorized to modify the base schemes. Moreover, changes must be done before generating the document information, otherwise the already-generated document will be changed accordingly.

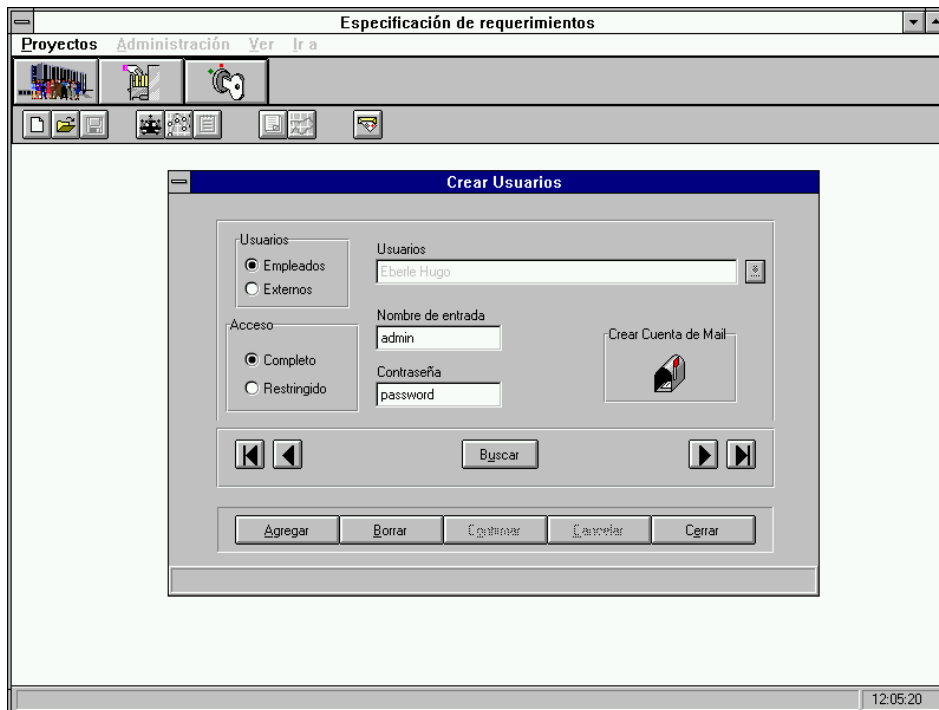


Figure 2

### ***Participants Management***

Participants include members of the organization and external users who use the tool. During the project definition, the project leader or manager must associate the participant/s with each of the stages, along with the role that they will have during the project. The different roles are defined in the methodology, and have already been explained. (figure 3)

As already said, and according to the definition in the JAD methodology, external



users may be participants and be a part of the team in some of the stages. They can approve, ask for changes, or reject documentation.

### Workflow management

The stages included in the methodology, already described, are organized to form the JAD circuit. Within each stage, the set of tasks included in it is defined. The JAD-CASE statically defines the path the tasks will follow for requirements specifications.

The same as with standard documentation, most projects match with the pre-determined base path. However, there may appear special cases which, generally due to their complexity, do not match with this base path. This may happen with medium and small projects which do not need to involve so many stages for their analysis. This is why workflow management allows to change the base path by removing some of these stages, those that, due to their characteristics, are not needed for the project under consideration. The manager does not allow the income of new tasks because it involves all the activities pre-determined by the JAD. (figure 4).

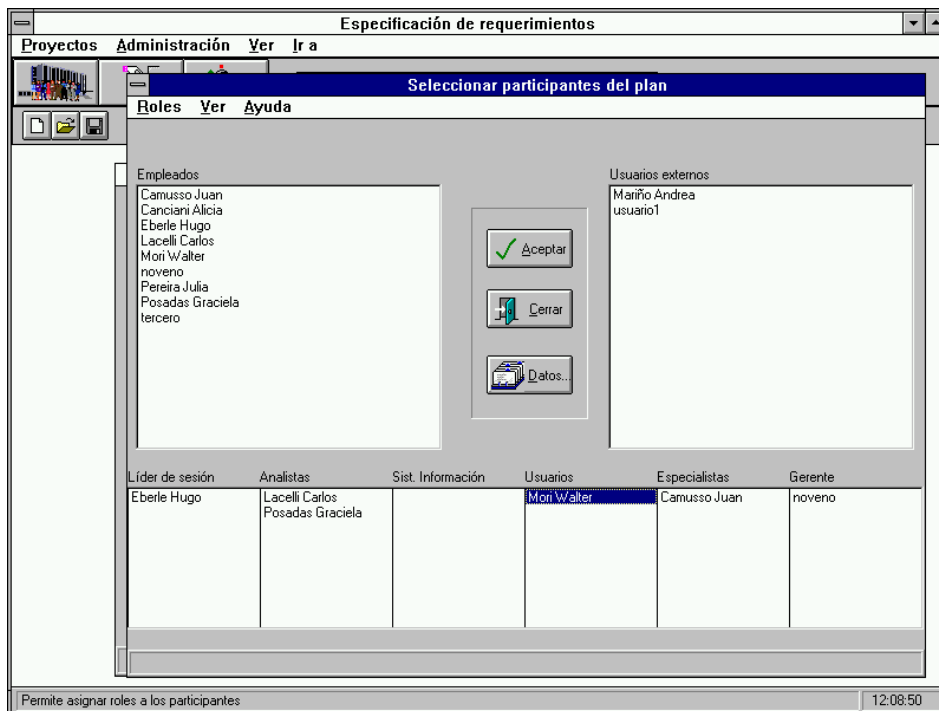


Figure 3

Each stage has its participants. In addition to this, each task within each stage has its executor/s<sup>9</sup>. One of these executors, in case there are more than one, is elected responsible for that task, and will be the one to inform that the task is finished, or will take responsibility in case there is a delay. The tool itself names a responsible in case there is no one assigned.

Each task is associated with a fixed duration, which shows the average time needed for the resolution of that task. Each project may vary that time, which means that a change in duration is considered. The unit chosen to measure this duration is days (figure 4). A proper definition of the duration is very important because it is used to follow deadlines, delays

<sup>9</sup> In charge of carrying out that task

analysis, and problems detection.

### Workflow

Stages are executed in the order pre-established by the methodology; and within each stage tasks also follow a certain order. For a stage to be able to begin, the previous one must be already finished.

In turn, the tasks within each stage respect a certain order and time; and they can be controlled the same as stages. Thus, a task can only be performed if it is enabled, and it will be enabled only if the previous tasks are already marked as finished. (figure 5)

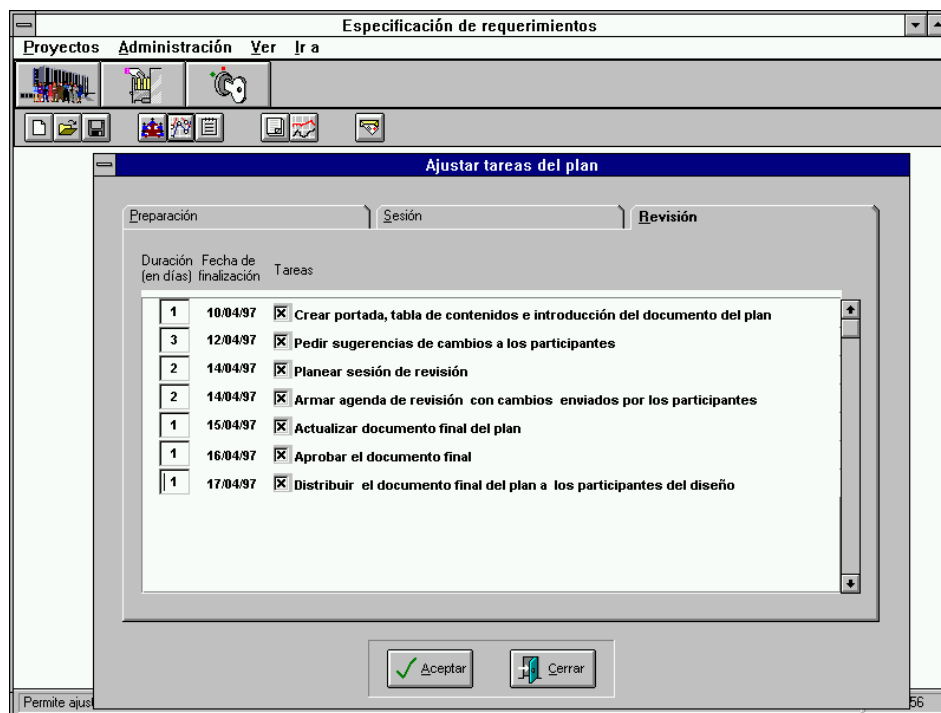


Figure 4

Each stage of the project, and each task in particular, may be in one out of five clearly identifiable states:

- *finished*: every stage or task already finished.
- *ready to start*: those stages or tasks which can start but have not been done yet.
- *pending*: stages or tasks that cannot be executed because at least one of their predecessors are not yet finished.
- *delayed*: stages or tasks which, according to the pre-established schedules (or eventually according to extensions of deadlines already agreed upon) are not finished yet.
- *under revision*: once finished, a task goes into revision when its "finished" mark is removed, in order to modify and/or correct it. When the task enters this revision state, first it makes the stage enter into that state, and secondly, it makes every other posterior task in the finished state go to the revision state as well. This state presents two instances:

- *ready to review*
- *pending*

which agree with the definitions above.

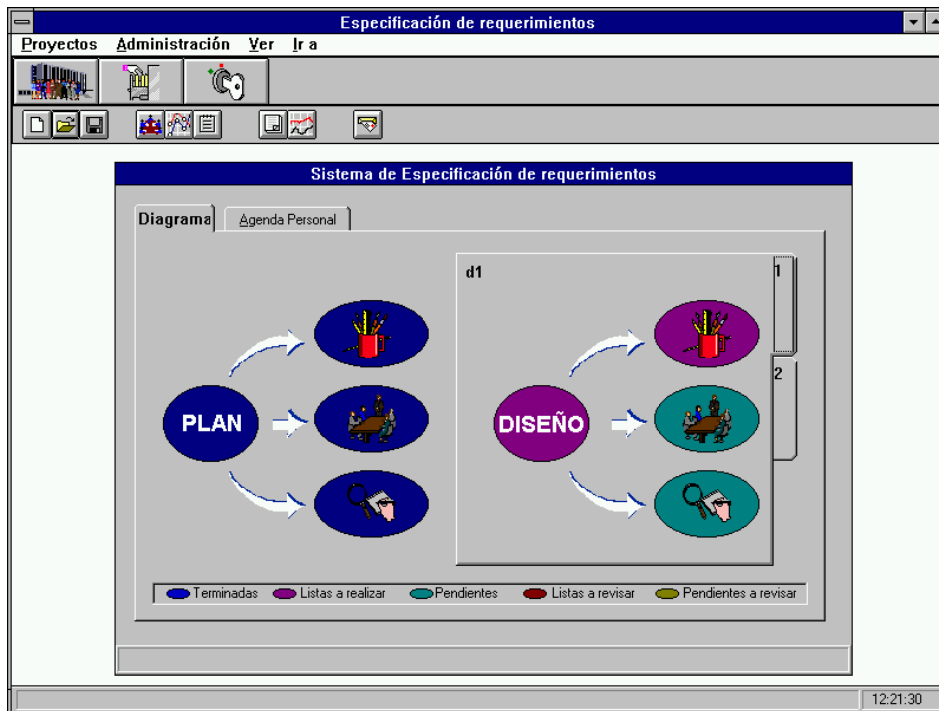


Figure 5

## Communication among participants

There are two levels of communication among participants:

- when a stage or a task is finished, every following one goes into the ready to start state. This action updates the state of the system by allowing participants in charge of the new ready-to-start tasks to be notified for that purpose.
- JAD-CASE includes a mail service.

## Security

For each project, the information generated using the tool is, in most of the cases, confidential. Therefore, security levels must be defined for its management. Security management is provided in two groups:

- within a project
- at an organization level

Within a project, each participant can only use and access the documentation corresponding to his role. At an organization level, certain participants may have access to:

- control all projects (tool manager)
- control all documentation of a project (project manager)

## Conclusions

The benefits of having a tool with the described characteristics are related to the following items:

- Documentation
  - Definition of the documentation most related to the reality of the problem being solved.
  - Availability of the previously generated documentation in a more organized fashion.
  - “On-Line” availability of the documentation, which allows to determine the stage of the requirements analysis.
  - Management of the documentation version.
- Methodology
  - Identification of each work group responsible.
  - Determination of the members of each work group, as well as the responsibilities assigned to each of them.
  - Determination of the tasks sequence.
  - Manual or automatic presentation of eventual modifications or changes.
  - Checking of the fulfillment of the different tasks of each phase.
  - Reduction of the communicational gap between user and analyst.
- Management
  - Realization of a project global control; that is, information about the project state is permanently available.

Although there are CASE tools in the market, which allow, among other possibilities, to analyze requirements, the objective of this work was to develop a product which would allow to evaluate the system requirements of the early Engineering stages, its critical point being communication with the user, by following the steps determined by the JAD methodology.

Even though the construction of the tool was experimental, it will be used at the LIDI to carry out system requirements evaluations, which should result in an increasing quality of the systems thus produced.

## Acknowledgment

To MS Alejandro Bianchi, professor of the Office Automation Master Course, for collaborating with his ideas to define and solve this work.

## Bibliography

[ALO95] *Exotica/FMDC: Handling Disconnected Clients in a Workflow Management System*. G. Alonso, R. Günthör, M. Kamath D. Agrawal, A. El Abbadi and C. Mohan, IBM Almaden Research Center, San Jose, CA, september, 1995.

[AUG91] *Joint Application Design* The group session approach to system Design, Judy

August, Yourdon Press Computing Series.

- [BAL97] *Una propuesta de workflow para la especificación de documentos para el proceso de Ingeniería de Requerimientos*. Balda, V.; Vicenzi, A.; Bertone, R. Anales del ICIE 97 (International Congress of Information Engineering - Buenos Aires - Argentina)
- [BAS91] *Developing Software for the User Interface*, Len Bass Joelle Coutaz, Addison Wesley, 1991.
- [BER96] *Workflow para la etapa de mantenimiento de software en el ciclo de vida clásico* (Ramón, Bertone, Russo, Gorga, Madoz) IV Encuentro Chileno de la Computación, ECC96.
- [BIA95] *JAD. Apuntes del Master en Automatización de Oficinas*. Alejandro Bianchi, UNLP, Facultad de Ciencias Exactas, Dpto. de Informática.
- [BRI93] *Modelling of Working Groups in Computer Supported Cooperative Work*. Stef Joosten and Sjaak Brinkkemper, Centre for Telematics and Information Technology, University of Twente, The Netherlands.
- [EDG95] *WORKFLOW: Applications Help Hone I/S Competitive Edge*. Jim Edgemon, Application Development Trends-Enterprise Client/Server Development, June 1995.
- [GEO94] *Workflow Management Concepts, Commercial Products, and Infrastructure for Supporting Reliable Workflow Application Processing*. Geogakopoulus, D. GTE Laboratories Incorporated
- [JOO95] *Fundamental Concepts for Workflow Automation in Practice*. Stef Joosten and Sjaak Brinkkemper, Centre for Telematics and Information Technology, University of Twente, The Netherlands.
- [MAY94] *Conozca las Herramientas de Flujo de Trabajo*. Thornton A. May, BYTE, Julio 94, Mc. Graw Hill.
- [MOH95] *An Overview of the Exotica Research Project on Workflow Management Systems*. G. Mohan, G. Alonso, R. Günthör, M. Kamath and B. Reinwald, IBM Almaden Research Center, San Jose, CA, september, 1995.
- [OLS91] *User-Centered Design of Collaboration Technology* G. Olson, J. Olson. Journal of Organizational Computing Vol 1, No 1. 1991.
- [PRE93] *Ingeniería del Software. Un Enfoque Práctico*. Roger S. Pressman, Tercera Edición, Mc. Graw Hill, 1993.
- [RAM96] *Herramienta para la administración de problemas y cambios*, (Ramón, Bertone, Gorga, Madoz, Perez Cerra) JUCSE 96, Santiago del Estero, Argentina.
- [WAL94] *Trabajo más organizado*. Scott Wallace, BYTE, Julio 94, Mc. Graw Hill.
- [WEB] *Workflow Management*. World Wide Web server of the Workflow Management Project at department of Computer Science of the University of Twente in the Netherlands. <http://www-is.cs.utwente.nl:8080/~joosten/workflow.html>

[YOU93] *Análisis Estructurado Moderno*. Edward Yourdon, Ed. Prentice Hall, 1993.

[YOU93] *Managing the System Life Cycle*, 2/e. Edward Yourdon, Yourdon Press Computing Series.