Semantic Based Support for Visualisation in Complex Collaborative Planning Environments

Natasha Correia Queiroz Lino

Doctor of Philosophy

Artificial Intelligence Applications Institute
School of Informatics
University of Edinburgh

2007

Abstract

Visualisation in intelligent planning systems [Ghallab et al., 2004] is a subject that has not been given much attention by researchers. Among the existing planning systems, some "well known planners" do not propose a solution for visualisation at all, while others only consider a single approach when this solution sometimes is not appropriate for every situation.

Thus, users cannot make the most of planning systems because they do not have appropriate support for interaction with them. This problem is further enhanced when considering mixed-initiative planning systems, where agents that are collaborating in the process have different backgrounds, are playing different roles in the process, have different capabilities and responsibilities, or are using different devices to interact and collaborate in the process.

To address this problem, we propose a general framework for visualisation in planning systems that will give support for a more appropriate visualisation mechanism. This framework is divided into two main parts: a knowledge representation aspect and a reasoning mechanism for multi-modality visualisation. The knowledge representation uses the concept of ontology to organise and model complex domain problems. The reasoning mechanism gives support to reasoning about the visualisation problem based on the knowledge bases available for a realistic collaborative planning environment, including agent preferences, device features, planning information, visualisation modalities, etc. The main result of the reasoning mechanism is an appropriate visualisation modality for each specific situation, which provides a better interaction among agents (software and human) in a collaborative planning environment.

The main contributions of this approach are: (1) it is a general and extensible framework for the problem of visualisation in planning systems, which enables the modelling of the domain from an information visualisation perspective; (2) it allows a tailored approach for visualisation of information in an AI collaborative planning environment; (3) its models can be used separately in other problems and domains; (4) it is based on real standards that enable easy communication and interoperability with other systems and services; and (5) it has a broad potential for its application on the Semantic Web.

Acknowledgements

I would like to gratefully acknowledge all the people and institutions which somehow influenced and supported the work documented here.

First I would like to acknowledge my research sponsor, CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, under grant number BEX 1944/00-2. A special thank you to Vanda Lucena and Yuri Lopes Zinn for being so helpful and efficient.

My biggest thank you to my supervisors, Austin Tate and Jessica Chen-Burger for all their continuous support, time dedicated, discussions, questions and comments.

A special thanks to my examiners Vladimir Geroimenko and Ewan Klein for their comments and suggestions and pleasant viva experience.

I wish to thank all members of the Centre for Intelligent Systems and their Applications (CISA) and the Artificial Intelligence Applications Institute (AIAI) for their direct and indirect influence in this work. A special thank-you to all involved in the AIAI I-X Project. In particular I would like to thank Clauirton Siebra, Jussi Stader, Eleanor Sim and Stephen Potter.

Thanks to the School of Informatics for the facilities and support provided, in particular the Informatics Graduate School (IGS) for support provided through the Research Travel Grants and the College of Science and Engineering for running the valuable Transferable Skills Programme. In addition a general thank to all administrative staff.

A thanks to all my office-mates that went with me through this journey, sharing not only the space but also experiences of research and life, including: Ana Costa Silva, Clauirton Siebra, David Lambert, DeLesley Hutchins, Eleanor Sim, Henrik Westerberg, João Cavalcanti, Laura Korte, Maciej Zurawski, Paul Martin, Richard Carter, Tom McCallum and Virgínia Brilhante.

I thank my past academic supervisors, Dr. Jacques Robin (Ph.D. in Computer Science, Columbia University) my Master's Degree supervisor and Dr. Ulrich Schiel (Ph.D. in Computer Science, Universität Stuttgart), my supervisor for the CNPq PIBIC (Programa Institucional de Bolsas de Iniciação Científica) Program.

A warm thank you to all my friends that supported me in pursuing this goal of finishing a Ph.D., either cheering me up, giving me the strengths to continue keep going, or understanding my absences. I can cite among them: Claudiana Freitas de França, Ana Karla Medeiros, Fernanda Ketchen, Gabriela Roca, Helik Rosa, Ana Costa Silva, Analy Teodoro, Flávia Bueno, Constantino Ribeiro, José Carlos Bins Filho, André Rauber Du Bois and Lazia Cade, etc. Also, my dear flatmates Rupert König and Kevin Hänni.

I would like to thank all my family for being a source of motivation and inspiration in life, however I have a big family and I cannot cite all here. So, a special thank you goes to my mother Tereza Queiroz, a professional role model for me; my grandfather Lauro Queiroz (*in memoriam*), the most dedicated and lovely person I have ever met, and a real father for me; my grandmother Avany Queiroz, a second mother; my youngest and dearest sister Paloma Silva; my uncle Lauro Queiroz Filho; my stepfather José Egesipe da Silva and my brother Rodrigo Freire. Also, I would like to acknowledge my special thank-you to my uncle Bruno Queiroz and aunt Fátima Camelo, the responsible people for introducing me to Computer Science in first place, still in the early ages of my life. Thank you all. Without you, it would not be possible.

Finally my thank you goes to Thomas Jüttemann, my guardian-angel, my special friend and my love; who supported me in the final and most difficult phases of my Ph.D., always giving me strengths, hope and kindness. I will always be grateful to you.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Natasha Correia Queiroz Lino*)

Table of Contents

List of Figures

List of Tables

Part I

Introduction and Motivation

Chapter 1

Introduction

The focus of this work is visualisation in planning systems. Visualisation is an aspect that is still not widely explored in intelligent planning systems (AI planning) [Ghallab et al., 2004]. Although many efforts have been made towards improving and developing new techniques and approaches for planning, they are centred in core planning problems, such as search algorithms efficiency, and very little work particularly addresses the problem of visualisation in AI planning.

The initial approach for AI planning, where a planner works in an isolated way, is giving space to the mixed-initiative style of planning where human agents play a role in the collaborative process of building plans. In this context, the existing lack of more elaborated approaches for visualisation in intelligent planning systems is compromising a broader application and use of such systems in real world problems. In real world situations, assisted planning services can be applied and supported by more sophisticated visualisation approaches. Based on these initial ideas, the contributions of this thesis have opted for investigating a broad and general solution, rather than choosing an approach only suitable to specific problems.

The remainder of this introduction is organised as follows. Section 1.1 first discusses the motivations and context of the work and gives a general outline of the thesis. Section 1.2 defines the problem scope, highlighting existing gaps in AI planning visualisation, and the need for more elaborated and general approaches to deal with visualisation of planning information. In addition, research opportunities in this area are pointed out through the identification of gaps. Finally the thesis structure is described in Section 1.3.

1.1 Motivation and Context

Despite Intelligent Planning being an area with a broad spectrum of research, involving investigation in both theoretical and practical subjects, there is a lack of research that focuses on the aspect of planning information visualisation. Among the existing planning systems, some well known and awarded planners do not even have a proposed solution for visualisation, and others only consider a specific approach when this solution sometimes is not appropriate for every situation.

The problem is increased when considering collaborative planning systems. With the transition from planners working in isolation in the past to today's mixed-initiative approach in AI planning, it is evident that there is a need for new forms of interaction between human and software planners. In such systems new requirements emerge since the agents that are collaborating in the process have different backgrounds, play different roles and have different capabilities, responsibilities, etc. The question is how will users make the most of planning systems if they do not have appropriate support for interaction with them? As such, there is a requirement in AI community to investigate planning from the perspective of information visualisation, while taking into account these new requirements.

From the AI planning point of view, depending on how it is approached, visualisation can play two main crucial roles in planning: (1) to permit collaboration among participant agents in the case of collaborative planning systems; (2) to allow proper interfacing between the software and human planners.

However, the existing lack of more generic and elaborated approaches compromises a broader application and use of such systems. Furthermore, it also compromises their application and use in real world problem domains and situations, where assisted planning services can be applied and supported by more sophisticated visualisation approaches.

So, in brief, the focus of this work is on the problem of visualisation in intelligent planning systems. The scope is delimited by the following main aspects: (1) a multi-modal visualisation approach enhanced by sound, (2) a context of collaborative environments of AI planning, where agents (human and software) work together to solve problems, and (3) use of mobile computing to support agents on the move.

We propose a general framework for visualisation in planning systems that will give support for an appropriate visualisation mechanism regarding the requirements we are considering. The essence of the general approach proposed is based on semantic modelling of the problem under the perspective of visualisation in AI planning. It consists of an integrated ontology set and a reasoning mechanism for multi-modal visualisation in contextual collaborative planning environments. The main idea is to give semantic-based support for visualisation in complex collaborative planning environments.

The framework is divided into two main parts: (1) a knowledge representation aspect and (2) a reasoning mechanism. In the knowledge representation aspect of this work, the ontology set will permit organising and modelling of complex problem domains from the visualisation perspective. The reasoning mechanism will give support for reasoning about the visualisation problem using the knowledge bases available for describing realistic collaborative planning environments.

In order to identify requirements for planning information visualisation in collaborative planning environments, a study was carried out about visualisation in AI planning systems. This study explored the state-of-art of the approaches most commonly used in AI planners for visualisation. In addition, some integrated scheduling systems were also analysed due to the similar nature of information that these systems present and manipulate.

This study permitted the identification of existing gaps in the area, such as the need for more elaborate and general approaches to deal with planning information visualisation. Furthermore, it also allowed the detection of many research opportunities in the area. For instance, one emerging opportunity is the integration of pervasive and ubiquitous computing to fill the gaps and support collaboration in real world domains. The integration of mobile and ubiquitous computing with artificial intelligence techniques has already been explored in recent years as it is surveyed in [Lino et al., 2003]. Such integration can add value to real world applications and fit the requirements of the scenarios we are dealing with.

In brief, the objective of this PhD thesis is the construction of a general framework for supporting information visualisation in AI planning. That framework intends to assess some of the main existing problems in the area. The main contributions of the approach proposed are: (1) it is a general approach for the problem of information visualisation in AI planning systems; (2) it will permit the modelling of the problem from the information visualisation perspective that will allow tailored support and reasoning about visualisation of collaborative planning information; (3) it is based on real standards to ease integration, communication and interoperability with other systems and services; (4) it has a broad potential for its application on the Semantic Web; (5)

in addition the framework will serve as a basis for implementations; and also (6) despite the models having been designed for contextual environments (collaborative AI planning), they are independent enough to be individually used for other application ends.

1.2 Problem Definition

The need for a broader use of knowledge-based planning has been discussed in recent years. In [Wilkins and desJardins, 2001] it is advocated that the use of knowledge-based planning will bring many advantages to the area, mainly when focusing on solving realistic planning problems. Complex domains can benefit from methods that use rich knowledge models. In this perspective, among the existing AI planning paradigms, *Hierarchical Task Network* (HTN)[Erol et al., 1994] is the one most appropriate to this proposition, in contrast to methods that use a minimal knowledge approach, for instance, the ones using a simple knowledge representation such as those based on STRIPS [Fikes and Nilsson, 1971]. However, despite the many advantages of the HTN paradigm, it also has limitations. Thus, there are many research opportunities in order to improve and permit a broader use of knowledge models in real world planning problems.

According to [Wilkins and desJardins, 2001] and based on their experience in planning for military and oil spill domains, the following capabilities are needed to solve realistic planning problems: (1) numerical reasoning, (2) concurrent actions, (3) context-dependent effects, (4) interaction with users, (5) execution monitoring, (6) replanning, and (7) scalability. However, the main challenges in real-world domains are that they cannot be completely modelled and consequently they raise issues about planner validation and correctness. Therefore, to make AI planning technology useful for realistic and complex problems, there is a need for improvement of the use of knowledge models in several aspects related to planning; and the development of methods and techniques that are able to process and understand these rich knowledge models.

Three types of planning knowledge are identified by [Kautz and Selman, 1998a]: (1) knowledge about the domain; (2) knowledge about good plans; and (3) explicit search-control knowledge. [Wilkins and desJardins, 2001] extended this list about planning knowledge mentioning that knowledge-based planners also deal with: (4) knowledge about interacting with the user; (5) knowledge about user's preferences; and (6) knowledge about plan repair during execution.

Recent researches are following these principles to develop more expressive knowledge models and techniques for AI planning. In [McCluskey and Simpson, 2004], for example, is proposed a work in this perspective of knowledge formulation for AI planning, in the sense that it provides support to knowledge acquisition and domain modelling through a system called GIPO (Graphical Interface for Planning with Objects). GIPO consists of a GUI and tools environment to support knowledge acquisition for planning. GIPO permits knowledge formulation of domains and description of planning problems within these domains. It can be used with a range of planning engines, since planners can input a domain model written in GIPO and translate into the planner's input language. GIPO has an internal representation, a structured formal language for the capture of classical and hierarchical HTN-like domains. Consequently, it is aimed at the classical and hierarchical domain model type. The advantages of GIPO are that it permits opportunities to identify and remove inconsistencies and inaccuracies in the developing domain model, and it guarantees that the domains are syntactically correct. It also uses predefined "design patterns", which are called *Generic Types*, that give a higher level of abstraction for domain modelling. To permit a successful use of AI planning paradigms, GIPO has an operator induction process, called *opmaker*, aimed at the knowledge engineer who may not have a good background in AI planning technology. However it assumes that they have knowledge about the domain. The GIPO plan visualiser tool allows engineers to graphically view the output of successful plans generated by integrated planners.

Based on these discussions of knowledge enrichment and broader use of knowledge-based planning, we argue that this vision should be even more augmented to other aspects. Our call is that knowledge enhancement can bring benefits to other areas related to planning, and we highlight the advantages that it can bring in the planning information visualisation area. That is the main focus of this thesis. We claim that knowledge models, developed from the AI planning information visualisation perspective, will permit semantic support and reasoning about the problem, that will come to fill some of the existing gaps in the area and open it to a broad diversity of other services.

Some of the existing gaps and problems that can be identified in the area of planning information visualisation are briefly introduced below (deeper discussions come later):

- Absence of solutions: many existing and awarded planning systems do not even have an approach for information visualisation proposed, such as the Graphplan [Blum and Furst, 1997] and Blackbox [Kautz and Selman, 1998b] planners;

- Lack of flexibility: the existing solutions for visualisation in planning systems, in general, adopt only one solution for presenting information, when, in some cases, it is not appropriate for every situation. The PRODIGY [Veloso at al., 1995] system for example, adopts only a GUI (Graphical User Interface) approach, while the TRAINS [Allen et al., 2001a] and TRIPS [Allen et al, 2001b] planners mainly use a natural language based solution (however map based solutions are also explored in these systems). Nevertheless, these solutions do not suit all different cases in real world domains of planning;

- Design for a specific aspect of the planning process: visualisation approaches used in AI planning systems sometimes do not give support to the entire planning process (including domain modelling, generation, collaboration, replanning and execution), but frequently, only to part of the process. There is a need to find general approaches to support planning information visualisation that will permit an uniform and integrated use of such approach for the development of solutions to every aspect of the planning process;

- Visualisation directly associated with the planning approach: information visualisation in AI planning systems sometimes is closely attached to the planning approach and related aspects, such as the domain of application, the paradigm or search algorithm for planning, the plan representation method, the plan product, integration to scheduling, etc. For instance, it is common in integrated planners and schedulers for an information visualisation solution to show temporal information, due to the nature of information that such systems manipulate. This limits the broad use and scope for interaction with other systems. Also, services that they can potentially provide are limited by the visualisation approach;

- The non-existence of general solutions: the issues discussed above make evident that there is a need of more global mechanisms that will provide general solutions for planning information visualisation. It is this gap that will be investigated in this thesis.

Having highlighted and discussed these problems and opportunities for research, we now describe the structure of the thesis.

1.3 Thesis Structure

This section describes the remainder of this document, summarising the thesis structure. Completing Part I, Chapter 2 introduces our approach to tackle the visualisation problem (a *Semantic Based Support for Visualisation in Complex Collaborative Planning Environments*). In addition, Chapter 3 exemplifies the approach in a motivational scenario.

Part II presents a bibliographic review of the main areas related to the background of this thesis: Chapter 4 covers the area of Information Visualisation in general, its main concepts and definitions, methods, techniques, etc. Chapter 5 makes an overview of information visualisation in AI planning systems. Some integrated scheduling systems were also included in this analysis. Chapter 6 brings our summary about this review, discussing in more detail the main problems and gaps in the area.

Part III presents our proposal. To that end, Chapter 7 discusses the semantic modelling approach, which consists in an integrated ontology set for describing planning information from a visualisation perspective. Chapter 8 gives attention to the reasoning mechanism, which uses knowledge about the domain (described via the ontology set) to infer modalities of visualisation to a plan or parts of it.

Part IV is about application scenarios, validations and conclusions, discussing for that a practical application of our framework, together with final remarks. Chapter 9 shows how the framework can be used in an application domain, based on a disaster relief operation, where several agents are carrying out different tasks in a collaborative environment. Finally, Chapter 10 discusses evaluation and the conclusions of this work, highlighting the contributions, problems and possible future directions.

Chapter 2

The Proposed Approach

This chapter introduces our approach for a general framework for information visualisation in AI planning systems. This general approach is based on semantic modelling and knowledge engineering techniques.

2.1 A General View

The approach presented in this thesis consists in the development of several semantic models that, when integrated, permit the modelling and expression of the problem of planning visualisation. The models support the construction of a reasoning mechanism for multi-modal information visualisation destined for use in collaborative planning environments.

The framework is divided in two main parts: a knowledge representation aspect and a reasoning mechanism. In the knowledge representation aspect, a set of ontologies allows the organisation and modelling of complex domains from the visualisation perspective. The reasoning mechanism, based on the knowledge base available and designed for realistic collaborative planning environments, allows a tailored support for information delivery and presentation, through reasoning about the visualisation problem.

The main aspects considered in the semantic modelling include: the nature of planning information and the appropriate tailored delivery and visualisation approaches for different situations; collaborative agents that are playing different roles when participating in the planning process; and the use of mobile computing and its devices diversity. This needs an appropriate approach with great expressiveness and flexibility.

The semantic model is composed of the following (sub) models: Visualisation

Modalities enhanced by Sound, Planning Information, Devices, Agents, and Environment. The next section presents these models in more detail, but here we give an introductory explanation:

- **Multi-Modal Visualisation Modalities Enhanced by Sound:** permits the expression of the different modalities of information visualisation considered in the approach, and in addition, includes sound as a relevant form to enhance cognition;

- **Planning Information:** represents planning information at a higher level of abstraction. It is based on <I-N-C-A> (Issues-Nodes-Constraints-Annotations) [Tate, 2001], the I-X Project ontology;

- **Devices:** this model permits descriptions of the features of devices in general. For example, mobile devices such as cell phones, PDAs, pocket computers, etc;

- **Agents:** allows the representation of agents organisations, including different aspects such as agents relationships (superiors, subordinates, peers, contacts, etc.), agents preferences, agents capabilities and authorities for performing activities, and also, agents mental states;

- **Environment:** this model allows the representation of information about the general scenario. For instance, position of agents and resources in terms of global positioning (GPS), etc.

Figure 2.1 illustrates the framework architecture. Through semantic modelling techniques (ontologies), several knowledge models complement each other to define a collaborative planning information visualisation scenario. These knowledge models permit the organisation and modelling of realistic collaborative environments of planning from an information visualisation perspective. Then a reasoning mechanism, applied to the knowledge bases available, results in outputs visualisation plans, tailored for each situation.

2.2 Visualisation Framework: Semantic Modelling

In the proposed approach, the definition of the Planning Visualisation Framework is expressed through five different models that define the main aspects of the problem. The next subsections will explain each of them in detail.

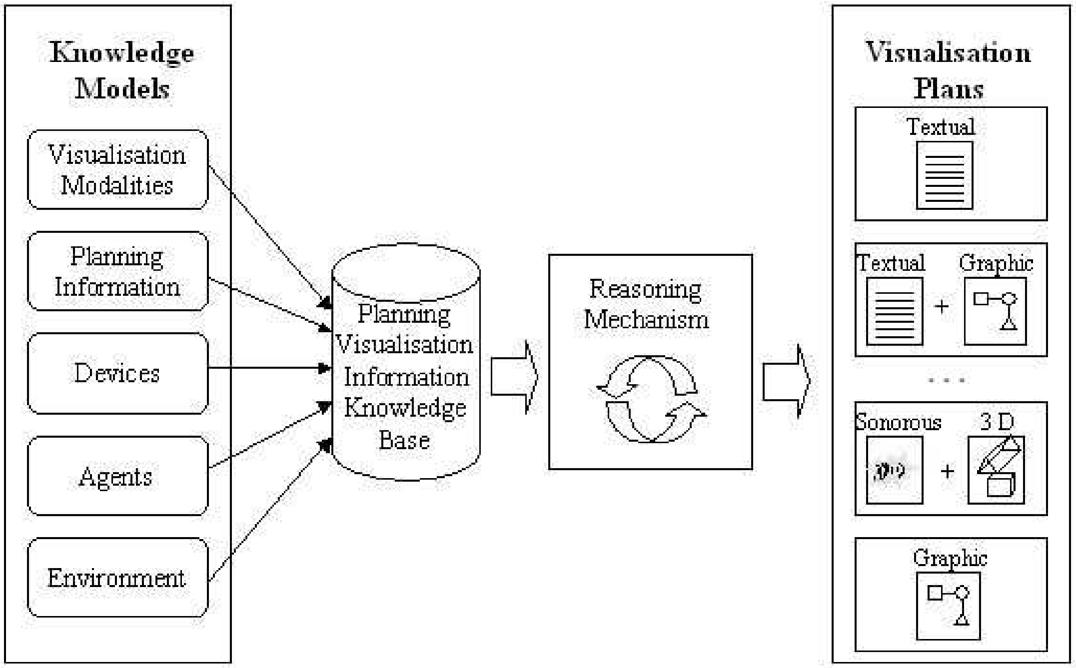


Figure 2.1: Framework architecture.

2.2.1 Multi-Modal Information Visualisation Ontology (Enhanced by Sound)

Information Visualisation (IV) is defined in [Card et al., 1999] as the use of computer-supported interactive visual representation of abstract data to amplify cognition. Many classifications of visual representation exist in the literature. [Shneiderman 2004] classifies data types of information visualisation in: 1-Dimensional, 2-Dimensional, 3-Dimensional, Multi-Dimensional (more then 3 dimensions), Temporal, Tree, and Network data. [Lohse et al., 1994] propose a structural classification of visual representations based on hierarchically structured categories. This classification is divided in six groups: graphs, tables, maps, diagrams, networks and icons. Another classification of visualisation types is proposed in [Burkhard 2004] from a perspective of architects. The visualisation types described are: sketch, diagram, image, object, and interactive visualisation.

These classifications are relevant in many aspects, including help to construct the framework categorisation, to understand how different types of visualisation communicate knowledge and identifying research needs. Furthermore, the existing development of prototypes for each category offers design guidance.

Related to user tasks in information visualisation, [Shneiderman 2004] classifies seven kinds: (1) *Overview* of the data set, (2) *Zoom in* on items, (3) *Filter out* items, (4) *Details-On-Demand* to select items and get details, (5) *Relate* to view relationships among items, (6) *History* to keep history of actions to support undo, replay, etc., and (7) *Extract* to allow extraction of subsets and of query parameters. Other tasks can be considered a special form of manipulation, such as *Direct Manipulation* or *Dynamic Queries* [Shneiderman, 1992] [Shneiderman 1994].

However, despite the power of information visualisation, in certain circumstances it is not sufficient to transmit knowledge to users. People assimilate information in different manners and have distinct limitations and requirements. For instance, deaf or hearing impaired people have different needs related to information acquisition. Therefore, different modalities of visualisation and interaction are needed for different users. For this reason, to permit broad possibilities of planning information delivery, included in the framework are not only visual representations but also other forms of user interaction, such as natural language interfacing that not only in textual form, but for instance also voice; sonification and use of sounds, etc., as other forms for communicating knowledge. To that end, the modalities and their concepts are modelled in the "Multi-Modal Information Visualisation Ontology", however enhanced by the aspect of sound. Sound was considered relevant to our work because it can play an important role in augmenting cognition in environments of collaborative planning. For example, in situation where human agents are carrying on plan execution on the move, they might need to use their hands and/or eyes to perform their tasks, so sound is a resource to be used in situations like that, to deliver information.

This model and ontology definition are derived from previous works on classifications of information visualisation [Wilkins and desJardins, 2001]. In addition, they are based on requirements for planning information visualisation to realistic problems, which is representative of the type of scenarios that are being targeted. The core of the semantic definition of this model is centred on multi-modal information visualisation and communication definitions and also on user tasks that can be performed upon the visualisation modalities.

The ontology includes the following main categories and concepts for a multi-modal approach of information visualisation, enhanced by sound:

- **1-D Textual:** based on textual representation of information. This modality is appropriated for simple devices that do not have many computational resources to present elaborated visual representations;

- **2-D Tabular/GUI/Map:** considers abstractions of information that are represented in two dimensions. For instance, tabular, GUI and map representation. Tabular defines a more structural way to present textual (but not only) information and together with GUI and map based, these representations require devices with more computational capabilities to present information;

- **3-D World:** considers three-dimensional representations of the world for information presentation. Due to the more sophisticated nature of the information structure, this category is suitable for more powerful devices;

- **Complex Structures:** includes complex abstractions of data representation for information visualisation, such as: Multi-Dimensional, Tree and Network representations. Multi-Dimensional concerns representations considering more then 3 dimensions. One example of abstractions of this type is the use of parallel coordinates [Macrofocus, 2005] that represent several dimensions via a vertical bar for each dimension. Tree and Network visualisation are also included in this category of complex structures. In the literature there are many approaches to address these structures and the nature of some data types can benefit from these forms of representation;

- **Temporal:** Many solutions for temporal data visualisation is proposed on the literature. Temporal data needs special treatment. For instance, works such as LifeLines [Alonso at al., 1998] address this problem. In the ontology, this modality abstracts the concepts involved in the presentation of temporal data;

- **Sonore (Audio/voice):** incorporates audio and voice solutions in the modelling. Audio and voice aids can be very useful in certain situations, where user agents are unable to make full use of visual information or doing manual operation of devices;

- **Natural Language:** natural language concepts are also considered within the semantic modelling. Although there are arguments which imply that natural language cannot completely substitute graphical interfaces [Shneiderman, 2000], it is suitable for many situations.

2.2.2 Planning Information Ontology

The "Planning Information Ontology" categorises, firstly at a high level, planning information about the following natures or aspects of planning:

- **Domain Modelling:** includes concepts of planning information related to domain modelling;
- **Generation:** here, the semantic modelling is concerned with plan generation information concepts and abstractions;
- **Execution:** includes vocabulary regarding plan execution;
- **Simulation:** models abstractions regarding plan simulation information.

Each of these aspects of planning deals with different types of information. Therefore, in this model/ontology a mapping was made trying to categorise types of planning information within each of the aspects of planning mentioned before, but keeping an information visualisation perspective in mind. Thus, in domain modelling, for example, we desire the visualisation of resources, environment, and/or goals definition. On the other hand, in planning generation we give more emphases to show the actions/operators applied to solve problems.

For the modelling of these ideas, the following concepts are considered in the ontology:

- **Planning Information:** the conceptual definition of planning information for the purpose of the visualisation framework is based on the <I-N-C-A> model [Tate, 2000] for collaborative planning processes. [Polyak and Tate, 1997] discussed comparisons among different planning representation languages as candidate for standards. The result of that analysis was that <I-N-OVA> [Tate, 2000], antecessor of the <I-N-C-A> model, had a better coverage rating in comparison to the other representation models. The study was made according to several rigorous process requirements (more details can be found in the paper), and that concluded that the <I-N-OVA> was the most general representation. Thus, the results of this study show that the <I-N-OVA>/<I-N-C-A> concepts fit the desired features of our approach;
- **Planning Information Aim:** considers that planning information can be used for different aims, these can be domain modelling, plan generation, plan execution and plan simulation. According to the literature and existing planning

systems, depending on the aim, planning information is approached in different ways. Thus, the framework recognises that information delivery for domain modelling is not the same as that for plan generation, for example;

- **Planning Information Delivery Strategies:** based on the literature and existing planning systems, it is possible to identify that each planning information aim category (domain modelling, plan generation, plan execution and plan simulation) deals, in general, with different types of information. As a result, different delivery strategy can be identified for each one, because there are different requirements of data presentation, summarisation, etc.

Therefore the main aim of this ontology is to abstract and model these concepts regarding planning information from the perspective of the general framework objective of information visualisation.

2.2.3 Devices Ontology

The devices ontology [Lino et al., 2004] permits the description of the types of devices being targeted, such as mobile devices, cell phones, PDAs, pocket computers, etc. The representation will be made in terms of their characteristics (device profiling): screen sizes, features, capabilities, etc. However, the representation is intended to be generic enough to permit easy extensions to future technologies. This is a positive aspect, mainly because the mobile computing area is evolving very fast.

Composite Capabilities/Preference Profiles (CC/PP) [W3 Consortium, 2004a] is an existing W3C standard for devices profiling. The approach of CC/PP has many benefits. First, it can serve as a basis to guide adaptation and content presentation. Second, from the knowledge representation point of view, since it is based on RDF (Resource Description Framework), it is a real standard and permits its integration to the concepts of the Semantic Web construction. For future works, we envisage a Semantic Web extension associated to this framework. Third, another advantage is that CC/PP provides resources for vocabulary extension, although extensibility is restricted.

On the other hand, CC/PP has some limitations when considering its application to the realistic collaborative planning environment that we are envisaging. It has limited expressive power that does not permit broader semantic expressiveness. Consequently it restricts reasoning possibilities. For example, using CC/PP it is possible to express that a particular device is Java enabled. However this knowledge only means that it is possible to run Java 2 Micro Edition (J2ME) on that device. But, it can have a broader meaning, for example, when considering "what it really means to be Java enabled?" or "what does J2ME support?". Answers for questions like these will permit a more powerful reasoning mechanism based on the knowledge available for the domain. For instance, if a device is Java enabled and if J2ME supports an API (Application Program Interface) for Java 3D, it is possible to consider delivering information in 3D models.

For that, there is a need to develop an improved model for devices profiling that will be semantically more powerful. It is necessary to incorporate in the model other elements that will permit enhanced knowledge representation and semantics. The "Devices Ontology" proposes a new model approach that intends to enhance semantics and expressiveness of existing profiling methods for computational devices, such as mobiles. Consequently, reasoning capabilities will also be enhanced.

Semantic improvement is categorised as follow in the new model being proposed:

- **Java Technology Semantic Enhancement:** this category intends to enhance semantics related to the Java world. It is not sufficient to know that a mobile device is Java (J2ME) enabled. On the other hand, providing more and detailed information about it can improve device's usability when reasoning about information presentation and visualisation on devices. For that, in this new proposed model is included semantics of information about features supported by J2ME, such as support for 3D graphics; J2ME APIs (Application Program Interface), for instance, the *Location API* that intends to enable the development of location-based applications; and also J2ME plug-ins, such as any Jabber [Muldowney and Landrum, 2000] plug-in that will add functionalities of instant messaging, exchange of presence or any other structured information based on XML.

- **Display + Sound + Navigation Semantic Enhancement:** one of the most crucial restrictions in the development of mobile device interfaces is the limited screen space to present information. Two common resources to bypass this problem are sound and navigation approaches. Sound has been used instead of text or graphics to present information. For example, by providing sound alerts that indicate a specific message to the user. Sound can be very useful in situations where users are on the move and not able to use hands and/or eyes depending on the task that they are executing. Navigation can also be used sometimes to improve user interface usability. However, good navigation design has some complexity due to: device diversity and because in some devices navigation is closely attached to the device's characteristics (special buttons, for example). So, this category intends to enhance semantics related to these aspects, providing good coordination and reasoning through these resources during collaborative processes.

- **Open Future New Technologies Semantic Enhancement:** this category of semantic enhancement is the more challenging one in this new model proposition. Mobile computing is an area that is developing very quickly. New devices and technologies are being created every day. In this way it's easy to create technologies that will be obsolete in few years time. Trying to overcome this problem, we envisage that it will be possible to provide semantics to future new technologies in mobile computing via general classes and vocabulary in the model and framework proposed.

2.2.4 Agents Ontology

This ontology is used to model and organise agents (software and human) regarding their mental states, capabilities, authorities, and preferences when participating in a collaborative process of planning. The development of this ontology is based on two existing concepts: BDI [Rao and Georgeff, 1995] and I-Space [Tate et al., 2002].

The main requirements of this model/ontology is to satisfy needs for reasoning about the roles of agents in the organisation when participating in collaborative processes of planning, and all aspects related to it. In addition, the agents mental states regarding their goals, strategies and preferences in the process are included.

BDI (Belief-Desire-Intention) [Rao and Georgeff, 1995] is the most popular concept used in agent-based modelling and programming. B stands for *Belief* (Data), D represents *Desire* (Goal) and I stands for *Intention* (Plan). Each agent has its own BDI instance so that to achieve some goal (Desire), the agent must analyse the related data (Beliefs) and choose an appropriate plan (Intention).

I-Space is the I-X concept for modelling collaborative organisations of agents. I-Space allows the management of organisational relationships such as peer/peer or superior/subordinate. For each of these relationships we can associate specific forms of interaction, which characterise each relationship in detail.

The following main concepts are modelled in the agents ontology:

- Mental States: represents the agents' mental states: Belief (B), Desire (D) and Intention (I);
- Roles: this concept is regarding the role the agent plays in the planning process. Roles are also associated with responsibilities, capabilities and authorities;
- Relationships: agents are organised in virtual organisations, such as hierarchical structures. Agents related to other agents, and these relationships can be: superior, subordinate, peer or contact. Relationships define some rules regarding, for example, delegation of tasks that has implications for information visualisation strategies;
- Preferences Profiling: through the concepts modelled here, agents can specify preferences regarding modality of visualisation, devices properties, etc. Based on these profiling techniques it is possible to adapt planning information presentation and delivery to the agent requirements.

2.2.5 Environment Ontology

The environment ontology is responsible for permitting expression of environment awareness. In particular, location-based awareness is being considered, where this type of information can be based on GPS (Global Positioning System), for example, and such like. Dealing with location-based information will allow the guidance of presentation of information. Therefore, the main concept modelled in this ontology is *Geographic Location*, where agents localisations in terms of global positioning and related properties are specified.

2.3 Reasoning Mechanism

This set of ontologies allows the development of reasoning mechanisms related to visualisation in collaborative planning environments. This section gives an example of reasoning considering device profiling.

As discussed previously, one of the goals of the knowledge models is to improve semantics. For instance, considering mobile computing, despite the existence of models for expressing concepts regarding device's profiles and features, they were not enough for the level of knowledge and reasoning we envisage. Thus, in our device ontology we tried to make available broader semantics that would permit improved reasoning for tailored information visualisation and delivery.

Figure 2.2 presents an extract of the devices ontology, using OWL (Web Ontology Language) as the knowledge representation language [W3 Consortium, 2005e], where we can see the definition of classes and properties that permit the Java question example of the previous paragraph be represented and used to reason upon. The class PDADevice allows the instantiation of individuals that represent a particular device. Through the JavaEnable property defined for this class, it is possible to express that a specific PDA is Java enabled. The unique instance of the J2ME class specifies the features of the J2ME platform. For instance, this class has the property 3DSupport that expresses the semantic of supporting features of 3D visualisation models or not.

Using the classes and properties defined in the devices ontology, it is possible to express instances of real world devices used by human agents in collaborative environments of planning. Hence, the reasoning mechanism uses the knowledge base and reasons upon it to tailor the delivery and visualisation of information.

An important question regarding the knowledge representation approach was deciding in whether to express the ontologies in OWL [W3 Consortium, 2005e] or RDF [W3 Consortium, 2005b]. One relevant aspect to consider was regarding semantic expressiveness, i.e., we wanted a language that would provide more ways of stating generalisations about the concepts involved, however in a formal way.

Using OWL rather than RDF was motivated by the fact that OWL provides additional vocabulary and also formal semantics to enhance semantic expressiveness and to facilitate machine interoperability. Both RDF and OWL have these language features: bounded lists, extensibility, formal semantics, inheritance, reification and inference. However there are some features only found in OWL, such as: cardinality constraints, class expressions, defined classes, enumerations, equivalence, local restrictions, and qualified constraints. These assumptions are valid for the languages specification until 2006. However in future versions of RDF and OWL languages specifications new features can be incorporated.

The meaning of these features are briefly discussed bellow, however a more complete explanation can be found in [Antoniou and Harmelen, 2004]:

- Bounded Lists: there is an indication that the list is complete;
- Extensibility: it is allowed new Properties to be used with existing Classes;
- Formal Semantics: it provides a formal notion of meaning that can be used for

automatic inference rules. Examples of techniques for specifying the semantics of a formal language are model-theoretic and axiomatic forms. A model-theoretic semantics provides a formal meaning for both RDF and OWL;

- Inheritance: RDF and OWL support *subClassOf* and *subPropertyOf* for inheritance definition;

- Reification: it provides a standard mechanism (for example, a statement to be the subject of another statement) for recording data sources, timestamps, etc. without intruding on the data model;

- Inference: OWL provides additional information useful for reasoning engines, such as, the constructs related to transitive, unambiguous, inverse of and disjoint properties. RDF/RDFS has basic support for reasoning based on class and property inheritance, however semantics is a prerequisite for reasoning support. The broader expressive power of OWL allows a richer inference support. Nevertheless, there is a trade-off between expressive power and efficient reasoning support. In general, the richer the language is, the less efficient the reasoning support becomes. So, a compromise is needed to guarantee computability;

- Cardinality Constraints: it limits the number of statements with the same subject and predicate (for instance *cardinality*, *minCardinality*, and *maxCardinality*);

- Class Expressions: it allows class expression, for example, in terms of union, disjunction, intersection and complement.

- Defined Classes: it allows new classes to be defined based on property values or other restrictions of an existing class or class expressions;

- Enumerations: it allows specification of a restricted set of values for a given attribute, for example, *oneOf*;

- Equivalence: it supports reasoning across ontologies and knowledge bases. For instance, *equivalentTo* can be applied for classes, properties, and instances;

- Local Restrictions: it allows restrictions to be associated with classes and properties. For example, associating domain and range with a property, allowing the color property to be used for different classes with different domains;

- Qualified Constraints: It permits expressions of qualified restrictions. There are examples of qualified constraints: *hasClassQ*, *cardinalityQ*, *minCardinalityQ*, and *maxCardinalityQ*.

Therefore, the absence of some of these features in RDF and the existence of others in the OWL language specification was a differentiator. Some of these features were necessary to our approach, to impose restrictions on the model/ontologies.

One aspect that RDF lacks, required for our approach, is an ontology language what can formally describe the meaning of terminology used in the models, but with more expressiveness. If machines are expected to perform useful reasoning tasks on these models, the language must go beyond the basic semantics of RDF and RDF Schema.

Nevertheless, what really differentiates OWL and RDF from our approach perspective is the common practice in the field of vision, sensor, mobile devices and/or planning. In these fields people are more inclined to use OWL rather than RDF. So it is a good practice to adopt a similar language, so that a translation between languages will not be necessary in case of our framework is adopted and integrated in other projects. Another argument is that our framework was designed to be extensible, thus, aiming at a language with more semantic expressiveness will easy this task of knowledge engineering and automatic processing, since OWL would offer more possibilities than RDF.

Also, another distinction between these two languages is the class level axiom descriptions presented in OWL, which allow one to operate at a class level. However we are not using this functionality in this first version of the framework, but that would be a good direction for further exploration.

In addition, we can say that RDF is more primitive as an ontological language and its reasoner is also not as powerful as the OWL one. A powerful reasoner is a desirable feature for us when examining and improving the ontologies/models. This facility is also necessary when the approach is used for different application domains.

An example of difference between RDF and OWL is: RDF enables that we assert facts, such as “agent X is named FireBrigade-1” or “location Y is a building in Kobe”. RDF Schema is more flexible, so that it enables that we describe vocabularies and create relations between them to describe things such as “agent X is in a location Y”. Differently, OWL enables that we describe relationships between vocabularies such as “fire brigades in *location* Y are in the same place that ambulances in *position* Z”. With OWL we can express that location is the same than position in our domain area.

Thus it exists in OWL semantic interoperability, and not only structural and syntactical interoperability as in RDF.

Related to the reasoning, using these relationship descriptions we can specify facts in our domain such as: (1) N-Dimensional is a modality, (2) both tree and network modality extend from N-Dimensional, (3) X is an agent, and (4) X does not have a tree library. Then if X must use a N-Dimensional modality, we could infer that X should use the network modality.

Another example is that using OWL descriptions we can specify, for instance, that the classes One-Dimensional, Two-Dimensional and Special-Structures are disjunctive. If we say that a device can only use the One-Dimensional modality, then the other two classes are automatically eliminated from the reasoning scope. Furthermore, all the modalities that extend these two classes are not taken in consideration during later reasonings (in this case only Sonore and Textual modalities are used).

Note that we are given here some introductory examples as motivation to justify the use of OWL rather then RDF as knowledge representation language in our approach. These examples are based on the conceptual semantic modelling, which is found in details in Chapter 7.

```
...
<!-- Information about the Ontology-->
<owl:Ontologyrdf:about="">
   <rdfs:comment>Devices Ontology</rdfs:comment>
</owl:Ontology>
...
<!-- Classes Definition -->
<owl:Class rdf:ID="PDADevice">
            <rdfs:comment>An instance of the class
               PDADevice represents the details of a
               particular device.</rdfs:comment>
            <rdfs:subClassOf
               rdf:resource="#CLDCConfigDevices"/>
</owl:Class>

<owl:Class rdf:ID="J2ME">
            <rdfs:comment>The instance of the class J2ME
                express the features of this
                platform.</rdfs:comment>
            <rdfs:subClassOf>
                        <owl:Restriction>
                                    <owl:cardinalityrdf:datatype=
&xsd;nonNegativeInteger">1</owl:cardinality>
                        </owl:Restriction>
            </rdfs:subClassOf>
</owl:Class>
...
<!-- Datatype Properties -->
<owl:DatatypePropertyrdf:ID="JavaEnable">
            <rdfs:domain rdf:resource=#PDADevice" />
            <rdfs:range rdf:resource="&xsd;boolean"/>
</owl:DatatypeProperty>

<!-- Datatype Properties -->
<owl:DatatypePropertyrdf:ID="3DSupport">
            <rdfs:domain rdf:resource=#J2ME" />
            <rdfs:range rdf:resource="&xsd;boolean"/>
</owl:DatatypeProperty>
...
```

Figure 2.2: Extract of the device ontology, part of our framework semantic modelling.

Chapter 3

Motivating Scenario

This chapter illustrates the framework proposed using a motivating scenario. The domain used for the scenario is the I-Rescue [Siebra and Tate, 2003], which suits the requirements for the types of domains we are envisaging. First, the I-Rescue scenario will be briefly introduced. Second, semantic modelling examples will be given using the ontologies that compose the framework and in addition reasoning cases are discussed. By examples, we will try to show that the reasoning component of the framework will permit adjustment of the visualisation modalities to several aspects related to the contextual collaborative scenario of planning: agents, devices, environment conditions and type of planning information requirements. The ontologies developed for conceptualisation and formalisation of such aspects play a role in facilitating reasoning. In this way, planning information will be delivered in a tailored manner.

3.1 Domain Characteristics

Despite the proposed framework being designed to be generic and domain independent, the type of domains we are envisaging applying the framework to would have the following characteristics:

- Realistic collaborative domains of planning;
- The complexity of the domain will include relevant planning knowledge to be modelled through the ontologies;
- Including both human and automatic input (human input is sometimes critical and automation can improve plan quality and reduce planning time);

- New situations can be created unexpectedly, resulting in a need to rapidly assemble responses; and

- Distinctive users participate in the process, thus it is important to have mechanisms to customize visualisation responses to suit the needs of a particular situation.

The domain we will use in this motivation chapter has these characteristics and will be introduced in the next sections. These characteristics will permit the exploration of the framework potential. The motivation examples presented here have the goal to give an introductory idea about how the framework works and motivate the reader, but later in this thesis it will be further explored with more robust examples.

3.2 I-Rescue Project

The framework is aimed at realistic domains of collaborative planning. The I-Rescue domain fits the requirements of such domains. The I-Rescue [Siebra and Tate, 2003] project is an effort to build knowledge-based tools to apply to search and rescue or disaster relief domains.

In I-Rescue scenarios, human and software agents work together and share knowledge and capabilities to solve mutual goals in a coalition support systems fashion. An important feature in systems like that is their ability to support collaborative activities of planning and execution. During planning processes, joint agents share knowledge so that a plan can be built in accordance with the perspectives of each agent. Then the activities in the plan execution are assigned to specific agents, which will use their individual capabilities to perform their allocated tasks.

I-Rescue scenarios consist of relief situations in natural disasters or adversities caused by humans. Situations like that need for an immediate response by joint forces with the main objectives of saving lives and minimising suffering. I-Rescue can be instantiated in many situations/scenarios.

The Kobe Earthquake of January 1995 is an example of how disasters have tragic effects in urban areas. More recently the tragedy of The Indian Ocean Tsunami in December 2004 that had unseen proportions of effects. Situations like that need an immediate response to relief human loss and suffering. The use of AI techniques and applications can help aid support, and a broad range of opportunities exist to do so.

We intend to contribute with our framework for information visualisation support. We are going to use the *Kobe Earthquake*, an instance also used by the I-Rescue project, for the purpose of scenario motivation for the framework proposed.

3.3 I-Kobe Domain Modelling

The *Kobe Earthquake* happened on Tuesday, January 17th 1995, at 5.46am (local time). It had the magnitude of 7.2 on the Richter scale. It is an example of how natural disasters can have an affect on human life. The Kobe region is the second most populated and industrialised area in Japan after Tokyo with a population of 10 million people. The earthquake shook the ground for 20 seconds, killing 5,000 people, and leaving 300,000 homeless. The estimated material damage was of ten trillion Japanese Yen (about 41,786,176.00 British Pounds) including roads, houses, factories and infrastructure.

Scientists are aware that Japan and other areas are more susceptible to earthquakes, due to the meeting of tectonic plates below the country's surface, and are studying ways to predict the occurrence of quakes more precisely. On the other hand, being aware of such predictions, computer scientists are also working on supporting technologies and tools to provide aids to disaster relief situations. The I-Rescue [Siebra and Tate, 2003] project, for example, is an effort from the AI (Artificial Intelligence) community to provide knowledge-based tools to aid search and rescue in disaster relief domains.

Based on this context, we have chosen the I-Kobe scenario for motivating our visualisation framework. We call *I-Kobe* a knowledge-based model inspired on the Kobe Earthquake. In the following, we are going to illustrate the modelling of the domain from the visualisation information perspective by means of the ontologies (Agents, Devices, Planning Information, Environments, and Multi-Modal Information Visualisation) proposed in the framework.

3.3.1 Agents

Different agents participate in the collaborative process of planning in the *I-Kobe* domain. Each agent has different characteristics such as: type, level of command, ability and quantity. Table 3.1 illustrates the modelled agents participating in the process.

The notation used here for agents names and functions is partially based on the one used in the *RoboCup Rescue Simulator*. The *RoboCup Rescue Simulator* (RCR)

Agent Type	Level of Command	Ability	Quantity
Search and Rescue Command Centre	Strategic	Command	1
Ambulance Centre	Operational	Coordination	1
Fire Station	Operational	Coordination	1
Police Office	Operational	Coordination	1
Fire Brigade	Tactical	Extinguish fire	2
Police Force	Tactical	Clear roads	2
Ambulance Team	Tactical	Rescue injured civilians	2

Table 3.1: Agents in the I-Kobe scenario.

[Kitano and Tadokoro, 2001] is a real-time distributed simulation system that is built of several modules connected through a network via a central *kernel*, which manages communications among such modules. The use of the RCR notation is due to the use of the simulator in other projects also related to the I-Rescue project, such as [Siebra and Tate, 2003].

3.3.2 Devices

Distinct devices can provide agents with information visualisation. Each device has different features, such as: mobility, screen size, processing capacity, networking and connectivity, etc.

Table 3.2 illustrates the type of devices considered for each of the agents illustrated in Table 3.1 on the I-Kobe domain:

In brief, agents that work on a strategic level (Search and Rescue Command Centre) are able to use more sophisticated computational resources, for instance, fully equipped command rooms. Agents that work on operational level (Ambulance Centre, Fire Station and Police Office) have access to desktop systems. Finally, agents working on a tactic level, normally working on the move, have access to a more restricted computational platform, making use of mobile devices.

Agent Type	Device Type
Search and Rescue Command Centre	Fully equipped command room
Ambulance Centre	Desktop
Fire Station	Desktop
Police Office	Desktop
Fire Brigade	Mobile device
Police Force	Mobile device
Ambulance Team	Mobile device

Table 3.2: Devices in the I-Kobe scenario.

3.3.3 Planning Information

Planning information can be classified according to different categories, for instance: Issues, Activities, Constraints and Annotations. However, depending on the planning process phase (generation or execution) the method/modality for dealing and visualising information can change. Table 3.3 illustrates examples of planning information to be dealt with by agents in the collaborative process of planning in the I-Kobe domain.

3.3.4 Environment

The environment ontology represents features of the environment that can have influence on the visualisation process. As discussed before, the principal component of this ontology is the agent location and its scenario of operation. For example, considering I-Kobe, the location of agents can be presented via a GPS approach if the agents are on the roads. However, if agents are performing inside of buildings, its position could be found out via the analysis of reflection signals from three different sources. Table 3.4 shows some environment elements that can influence the visualisation process.

We can imagine several other domains where the environment has an influence on the information delivery process. For example, during space operations, GPS and maps are not appropriate, for underwater missions in general humans cannot talk because of their breathing equipment, and scenarios that involve illumination issues. For this last case, consider the situation where agents are taking part in some nocturnal military mission, where the use of bright devices could expose their positions. Thus, a sound-based system could be a better solution in situations like that.

Planning information	Description
Issues	Outstanding questions to be handled and can represent unsatisfied objectives or questions raised as a result of analysis or other deliberative processes.
Activities	Represent components that are to be included in a plan. They can themselves be plans that can have their own structure with sub-activities and other elements.
Constraints	Restrict the relationships between activities to describe only those plans within the plan space that meet the requirements. Constraints have an associated type such as world-state or temporal.
Annotations	Account for adding complementary human-centric and rationale information to plans. They can be seen as notes on plan components, describing information that is not easily represented via the previous components.

Table 3.3: Information used during planning generation and execution.

3.3.5 Multi-Modal Information Visualisation

Finally, Table 3.5 shows preferences about visualisation methods, according to the multi-modal information visualisation ontology. The table is a simple example of how different agents of the I-Kobe scenario (Commander, fire station and fire brigade) can have different visualisations preferences based on their current planning aspect (generation or execution). Note that depending on the planning aspect and agent role, agents are performing different tasks (fire prediction, monitoring, etc.). These tasks manipulate a different set of information, which can require different methods of visualisation. The examples in the next section (Section 3.4) stress these ideas, showing how all this knowledge, modelled via ontologies, can be used to reach an appropriate information delivery method.

Note that the information in parentheses indicate the visualisation methods suggested, where: MAP stands for the map visualisation method, TEXT the text mode, GUI for the graphical user interface modality, while NLP stands for natural language processing types of modalities. A detailed overview of the visualisation modalities are presented in Chapter 7.

Element	Example in I-Kobe
Space	Indoor (in buildings), Outdoor (on streets)
Position	GPS (plain scenarios like streets), triangulation-based (in buildings or other scenarios that require a better location precision)
Illumination	Normal (daytime operations), restricted lighting (nocturnal operations or in tunnels)

Table 3.4: Examples of environment features.

	Plan generation	Plan execution
Commander	Fire prediction (MAP)	Information acquirement (TEXT)
Fire Station	Schedule (GUI)	Monitoring (GUI)
Fire Brigade	Pathfinder (MAP)	Report generation (NLP/voice)

Table 3.5: Examples of categories of visualisation.

3.4 Reasoning Examples

The reasoning mechanism is based on scenarios. A **scenario** is defined by instance inputs of the models/onlotogies (Agents, Planning Information, Devices, Environments and Multi-Modal Information Visualisation). Figure 3.1 illustrates the basis of how the reasoning works in the framework. First, instances of the models/ontologies feed a knowledge base. Following, these instances define different contextual scenarios. The reasoning then occurs upon these defined scenarios. The reasoning is based on a set of simple rules. The reasoning results in the delivery of tailored visualisation modalities, suitable to each specific scenario.

The next sections will discuss three reasoning motivation examples, each trying to focus on a different aspect of reasoning. The reasoning example in Scenario 1 is agent-oriented, the reasoning example in Scenario 2 is resources-oriented, while the reasoning example in Scenario 3 is device oriented. In each scenario, the reasoning mechanism will use rules that will determine a tailored visualisation approach suitable to the situation.

3.4.1 Reasoning Example Scenario 1 - Agent-Oriented Rules

Scenario 1 is defined by a *Command Centre* agent, participating in the generation phase of the planning process and using a fully equipped command room to visualise plan-

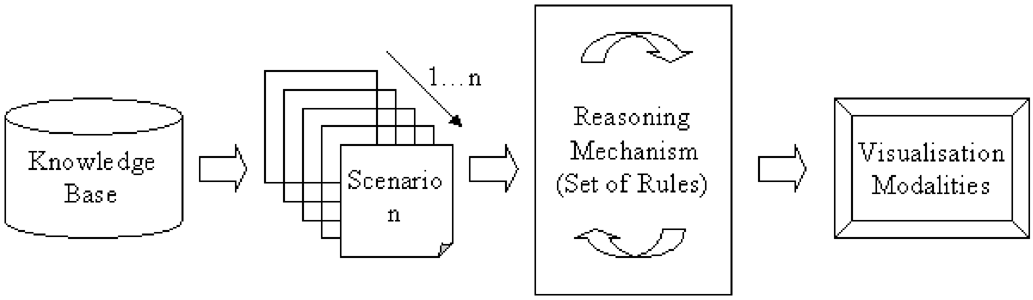


Figure 3.1: Reasoning mechanism.

ning information. A command centre agent performs at the strategic level of decision-making, which accounts for developing plans at a high level of abstraction. Thus, the principal tasks of this agent are related to analysis of information and definition of directions and priorities.

In the I-Kobe disaster relief domain, where several fires are spreading over the city, the planning model can contain the task "Control fire-spreading". So, the command room function, in this case, is to analyse the situation, make predictions according to the information available and decide where to concentrate the resources to avoid the fire spreading. For that, agents will need to access and manipulate world state information such as: position of fires, speed and direction of wind.

Based on this scenario definition, the reasoning mechanism will apply a certain group of rules to determine a suitable information visualisation approach. This set of rules determines if the information visualisation approach is mainly agent-oriented, device-oriented, planning information oriented or environment oriented.

Scenario 1 is an example where agent-oriented rules should be applicable. This is due to many aspects, but one argument is that since command rooms are fully equipped with resourceful devices for planning information visualisation, it is pointless, for example, to reason about visualisation in devices with restrictions. In this case, it is more important to reason according to the agent's characteristics: roles, preferences, abilities, etc. A set of rules that can be applied to this case is expressed in follow:

D_{device} = (command-room,desktop,mobile)

$D_{visualisation}$ = (text,GUI,map-based,sound,voice,3D,NPL)

D_{plan} = (generation,execution)

$\forall x,y\ x \in D_{device} \wedge y \in D_{visualisation} \wedge$ Is(x,command-room) $\Rightarrow$ Possible(y)

$\forall$x,y x $\in$ D_{plan} $\wedge$ y $\in$ D_{agent} $\wedge$ Is(y,commander) $\Rightarrow$ RequiredInf(world-state)
$\forall$x x $\in$ D_{plan} $\wedge$ Is(x,generation) $\wedge$ RequiredInf(world-state) $\Rightarrow$ Pref(map-based)

The first three statements restrict the domain of devices, visualisation methods and plan phases to a discrete and simple set of elements. Note that in a real domain, such sets tend to be much more complex. For example, the element mobile in the device domain can be represented by several different PDA device models. The next three lines (rules) express the following ideas respectively:

- If the device used for visualisation is a command room, then any kind of visualisation is possible because this device has a broad range of computational capabilities and high processing power;
- For any planning phase, the principal information required by the commander is related to states of world;
- During the plan generation phase, if the required information is related to the state of the world, then the preference for visualisation is the map-based method.

Then, a possible output of the visualisation reasoning system to represent the world state properties, in specific the wind properties, during the planning generation to the commander agent could be represented by the figure below (Figure 3.2):

3.4.2 Reasoning Example Scenario 2 - Planning-Oriented Rules

Scenario 2 is defined by a *Fire Station* agent, participating in both plan generation and execution phases. Its device is a common desktop, which we are assuming has limited processing capabilities to run 3D and NPL applications. A fire station agent performs at the operational level of decision-making, which accounts for refining the plans produced at the strategic level, mainly providing the logistical resources for them via processes of resource scheduling and load balancing.

The plan generation of fire stations can be summarised in the following way. When the fire station receives objectives from the strategic level, it starts by checking the necessary conditions and options to reach the objectives, according to their available resources that are represented by fire brigades. Using a scheduling technique, the fire station can choose the best configuration to allocate tasks to their fire brigades so that resources and time are elements that must be represented together in this planning phase.

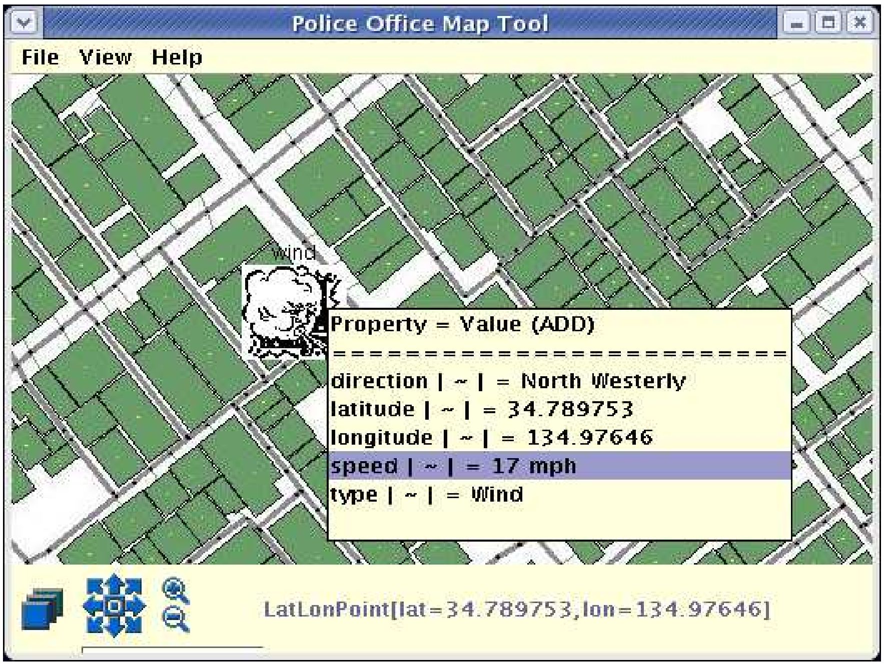


Figure 3.2: Example of visualisation output for scenario 1.

During the phase of plan execution, fire stations perform the task of monitoring the performance of their subordinates to check the status (ready, executing, complete or impossible) of the delegated activities. So we can note that the kind of information manipulated here is different from the plan generation phase.

Intuitively, lets consider that both types of planning information (resource/time and reports) are better visualised via a GUI-based visualisation. In this way, we can write the following rules for this scenario:

$$\forall x,y\ x \in D_{device} \wedge y \in D_{visualisation} \wedge Is(x,desktop) \wedge \neg(Is(y,3D) \vee Is(y,NLP)) \Rightarrow Possible(y)$$

$$\forall x,y\ x \in D_{plan} \wedge y \in D_{agent} \wedge Is(y,FireStation) \Rightarrow Preference(GUI)$$

While the first rule synthesises the idea that a desktop device is able to run any option from the visualisation domain, apart from NLP and 3D options, the second rule says that independently of the plan phase, the preferential visualisation method to fire brigades is GUI.

This scenario uses some simplifications so that the real complexity of the visualisation problem is hidden. For example, there are several types of desktops so that some of them are able to run 3D and NLP applications, while others are not able to do that.

In this way, the reasoning process must verify details of each device, which must be specified in some form of knowledge representation. Another interesting example is the use of graphics (GUI) as a form of visualisation. The second rule implicity says that resource/temporal and reports planning information are better visualised via GUI. However, in fact, the GUI for these two kinds of information can be very different. For example, to the first set of information (temporal/resources) the GUI tends to be similar to Figure 5.11 (Chapter 5), while the GUI for reports could be based on the idea of Figure 3.3, where colours represent the status of the activities (White: not ready for execution; Orange: ready for execution, Green: in execution; Blue: execution completed).

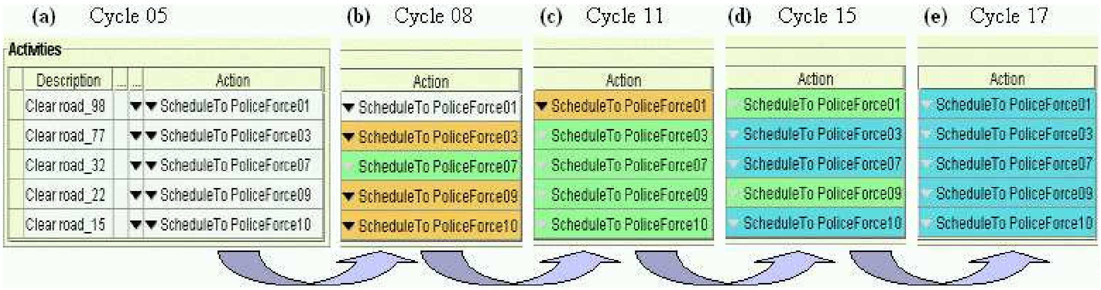


Figure 3.3: Example of visualisation output for scenario 2.

3.4.3 Reasoning Example Scenario 3 - Device-Oriented Rules

Scenario 3 is defined by an *Ambulance Team* agent, performing the activity of rescuing an injured civilian in a collapsed building. Since the ambulance is an agent on the move, it makes use of a mobile device to visualise information. This device has several limitations so that the range of visualisation methods is very restricted.

Ambulance teams can use, for example, a pathfinder that looks for the best routes to specific destinations, or a patrolling mechanism to trace routes that efficiently cover search areas. Such mechanisms are used during the plan generation phase of the ambulance team and both mainly require information about the world state (e.g., clear roads) to perform their tasks. A set of rules to describe the visualisation method for this scenario can be written as:

$$\forall x\, y \in D_{device} \wedge y \in D_{visualisation} \wedge \text{Is}(x,\text{mobile}) \wedge \neg(\text{Is}(y,\text{3D}) \vee \text{Is}(y,\text{NLP}) \vee \text{Is}(y,\text{GUI})) \Rightarrow \text{Possible}(y)$$

$$\forall x \in D_{plan} \wedge \text{Is}(x,\text{generation}) \wedge \text{RequiredInf}(\text{world-state}) \wedge \text{Environment}(\text{outdoor})$$

$$\Rightarrow \text{Preference(map-based)}$$

$$\forall x \in D_{plan} \wedge \text{Is(x,generation)} \wedge \text{RequiredInf(world-state)} \wedge \text{Environment(outdoor)}$$
$$\Rightarrow \text{Preference(3D)}$$

This first rule states the kind of visualisation mechanisms (3D, NPL and GUI) that are not supported by the mobile device. The second and third rules specify the context in which a map (Figure 3.4) and a 3D visualisation are the best choice respectively. These two rules also show how features of the environment, in this case outdoor/indoor, have a role in the decision process. For that, the model proposed needs to have a way to express the features about the environment.

Note that all the visualisation methods are represented here as preferences. However, the knowledge base needs to have rules to deal with cases where the visualisation preferences are not able to be applied. For example, the third rule says that a 3D is the best option if the operation environment is indoor. However the first rule says that the mobile devices do not support this method of visualisation. Thus, the reasoning mechanism must be able to find other visualisation methods for this situation.

Figure 3.4: Example of visualisation output for scenario 3.

Part II

Information Visualisation in Intelligent Planning Systems and Schedulers

Chapter 4

Information Visualisation

The main aim of Part II of this thesis is to give an overview about the use of information visualisation in intelligent planning systems. For that end, Part II is divided in three chapters. Chapter 4 first introduces some basic concepts and definitions of the field of Information Visualisation (IV), which are necessary for better comprehension and are subsequently used as basis for other chapters. Then, in Chapter 5, Information Visualisation (IV) is analysed within in the scope of intelligent planning systems. Finally, Chapter 6 discusses the main problems and gaps in the area of information visualisation in planning systems, together with the identification of research opportunities.

Information visualisation is an important area of intelligent planning systems, however it is still not very well explored and investigated. Trying to understand the area and its problems and gaps, these three next chapters mainly make an analysis of information visualization in Artificial Intelligence (AI) planning systems. This study intends to define how visualisation approaches are characterised in AI planning systems. Some of the questions that are addressed are: what kind of information AI planning systems manipulate, have as input, and present as output; which aspects of a planning process need to be interfaced with users; and which are the main types of approaches that the systems adopt to interface with users and for information visulisation.

4.1 Basic Concepts and Definitions in Information Visualisation

4.1.1 Definition and Origin of the Field

Visualisation itself is defined in [Card et al., 1999] as the use of computer-supported interactive visual representation of data to amplify cognition, where cognition can be defined as the acquisition or use of knowledge to permit insights.

Visualisation is originally a subfield in the area of Scientific Computing. Visualisation in Scientific Computing [McCormick and DeFanti, 1987] is concerned about handling large sets of scientific data and to enhance scientists' ability to see phenomena in the data. In this field data tends (but it is not necessary) to be based on physical data (human body, earth, molecules, etc.), where the computer is used to make visible some properties. Despite abstract visualisation being produced in this field, the information in inherently geometrical, based on physical space, for example, the visualisation of ozone concentration in the atmosphere.

Information Visualisation is a different field that tries to incorporate the realm of abstraction. This field is motivated by three main issues: (1) how to cast nonphysical information in a visual form, such as financial data, abstract conceptions, etc.; (2) how to render visible properties of the objects of interest; and, (3) since this kind of information does not have any obvious spatial mapping, there is the problem of mapping nonspatial abstractions into effective visual form.

Information Visualisation [Card et al., 1999] is then defined as the use of computer-supported, interactive visual representations of abstract data to amplify cognition. Visual aids to cognition benefit from good visual representations of a problem and from interactive manipulation of those representations.

In the last decades, Information Visualisation passed from being a new research field into the mainstream of user interface and application design. Several factors influenced this development, however the development of new and more powerful graphic hardware was a decisive one. First the Silicon Graphics workstation and its competitors in the mid eighties (1980) permitted the development of real-time interactive graphics for animation, geometric transformation in 2D and 3D, new visual effects, and allowed exploration of visualisation techniques for abstract information. Later, in the nineties (1990), the absorption of these graphics capabilities into the standard PC computer platform allowed information visualisation to be used in mass-market products.

The following concepts are related to Information Visualisation [Card et al., 1999]:

- **External Cognition**: interaction of cognitive representations to support thinking;
- **Information Design**: design of external representations to amplify cognition;
- **Data Graphics**: use of abstract, visual representations of data to amplify cognition;
- **Visualisation**: use of computer-based, interactive visual representations of data to amplify cognition;
- **Scientific Visualisation**: use of interactive visual representations of scientific data, typically physically based, to amplify cognition; and finally,
- **Information Visualisation**: use of interactive visual representations of abstract, nonphysically based data to amplify cognition.

Historically several fields originated the one that today is called Information Visualisation, such as:

- **Data Graphics**: work in data graphics, for example Playfair (1786) was among the earliest to use abstract visual properties such as line and area to represent data virtually [Tufte, 1983]. Tufte also published a theory of data graphics that emphasized maximizing the density of useful information;
- **Cartography**: in 1967 Bertin published a theory of graphics called *The Semiology of Graphics* [Bertin, 1967/1983] that identified the basic elements of diagrams and designed a framework for their design;
- **Exploratory Data Analysis**: the data graphics community was always concerned with statistical graphics. However, in 1977 Turkey began a movement from within statistics with his work on Exploratory Data Analysis [Tukey, 1977], where the emphasis was not on the quality of the graphics, but on the use of pictures to give rapid statistical insight into data.

The first IEEE Visualisation Conference was held in 1990, led by a community of Earth resource scientists, physicists, and computer scientists in supercomputing. Information Visualisaton was used as a method to accelerate analysis, and to enhance identification of interesting phenomena of data coming from satellites. It was also seen as an useful to replace expensive experiments by computational simulation.

At the same time, there was also interest by computer graphics and artificial intelligence communities in automatic presentation, the automatic design of the visual presentation of data. Mackinlay's thesis APT [Mackinlay, 1986], formalized Bertin's [Bertin, 1967/1983] design theory, added psychophysical data and used it to generate presentations. Other examples are Roth and Mattis [Roth and Mattis, 1990] who built a system to do more complex visualisation and Casner [Casner, 1991] who added a representation of tasks. However, the concern in this community was not on the quality of graphics, but on automating the matching between data types, communication intent, and graphical representations of the data.

Finally, the user interface community saw advances in graphics hardware opening the possibility of a new generation of user interfaces. These interfaces focused on user interaction with large amounts of information, such as multivariate databases or document collections.

The first use of the term *Information Visualisation* was in Robertson, Card and Mackinlay [Robertson et al., 1989]. Feiner and Beshers [Feiner and Beshers, 1990] presented a method for showing six-dimensional financial data in immersive virtual reality. Shneiderman [Shneiderman, 1992] developed a technique called dynamic queries for interactively selecting subsets of data items and tree maps, a space filling representation for trees. Card, Robertson and Mackinlay [Card et al., 1991] presented ways of using animation and distortion to interact with large data sets in a system called the Information Visualizer. The concern was the means for cognitive amplification not graphic quality, and interactivity and animation were important features of these systems.

All these communities that originated and influenced what today is the field of *Information Visualisation* mutually influenced each other and were followed by refinements and new visualisations.

In [Card et al., 1999] are given examples of information visualisation methods:

- **Active Diagrams**: amplifies the effect of a good visual representation by making it interactive;

- **Large Scale Data Monitoring**: uses information visualization to monitor and make sense of large amounts of dynamic, real-time data. It can be classified also as support decision visualisations;

- **Information Chromatography**: very abstract visualization of real time data to detect complex new patterns in very large amounts of data , such as the detection of telephone fraud.

In addition, there is also defined in the literature a more general concept than visualisation, which is called *perceptualisation*. Perceptualisation can be supported not only by visualisation but also sonification and tactilisation of data. However it is claimed by most authors that vision is the sense with the largest bandwidth.

4.1.2 Information Visualisation and Cognition

Information Visualisation [Card et al., 1999] has been defined as the use of computer-supported, interactive visual representations of abstract data to amplify cognition. The psychology definition for cognition is the action or process of acquiring knowledge and understanding through experience or the senses [Soanes and Stevenson, 2004].

Information Visualisation can, for example, support these cognition processes in the stages of a knowledge crystallisation task [Card et al., 1999]. A knowledge crystallisation task is one in which a person gathers information (data) for some purpose, makes use of it by constructing a representational framework (a schema) and then packages it into some form for communication or action. The results can be a briefing, a short paper, a decision or action.

Some of the characteristics of a knowledge crystallisation task are: (1) use of large amounts of heterogeneous information; (2) ill-structured problem solving; and (3) relatively well-defined goal requiring insight into information relative to some purpose.

Tasks of these kinds motivate attempts to develop information visualization. A typical scenario for a knowledge crystallisation task has the following elements:

- Information foraging;
- Search for schema (representation);
- Instantiate schema with data;
- Problem solve to trade-off features;
- Search for a new schema that reduces the problem to a simple trade-off;
- Package the patterns found in some output product.

In a scenario like that, Information Visualisation can aid the process of producing patterns that can be detected and abstracted. This process of abstraction is a fundamental principle for reducing the amount of information to a degree that can be processed by humans to give acceptable response to a changing of environmental circumstances. Information Visualisation can be applied to most phases of knowledge crystallisation.

Larkin and Simon [Larkin and Simon, 1987] have a study that illustrates how visualisation can be effective in the process of amplifying cognition. This study compares solving physics problems using diagrams versus using non-diagrammatic representations. More specifically, they compared the effort that had to be expended to do search, recognition, and inference with or without the diagram. The conclusion of the study was that the diagrams helped in three ways: (1) reducing search by grouping together information that is used together; (2) reducing search and working memory by avoiding the need to match symbolic labels using location to group information about a single element; and (3) automatically supporting perceptual inferences that were extremely easy for humans via the visual representation. To summarize, these ways improve the calculation of the function for accessing information and reduce the cost of certain operations. To understand the effectiveness of information visualisation, it is necessary to understand what it does to the cost structure of a task. Cost structure of information is a kind of information cost landscape. More details on cost structure can be found in [Shneiderman 2004].

[Shneiderman 2004] proposes six ways in which information visualisation can amplify cognition, however depending on appropriate mapping of information into a visual form:

- Increasing resources;
- Reducing search;
- Enhancing recognition of patterns;
- Perceptual inferencing;
- Perceptual monitoring; and
- Manipulable medium.

In [Kerpedjiev et al., 1998] is discussed a methodology for stating intentions in graphical form. The methodology proposed consists of an automatic realization of communicative goals in graphics. The approach is based on a task model that mediates the communicative intent and the selection of graphical techniques. The methodology has the following functions:

- Isolate assertions presentable in graphics;
- Map such assertions into tasks for the potential reader; and
- Select graphical techniques that support those tasks.

They presented a study case consisting in a redesigning of a textual argument into a multimedia one, applying graphics to achieve some of the intentions.

4.1.3 Data Treatment and Presentation in a Visual Form

In order to provide a suitable visualisation of information for the human perceiver, it is necessary to have a series of mappings from raw data to visual form. Figure 4.1, from [Shneiderman 2004], shows a diagram of these mappings.

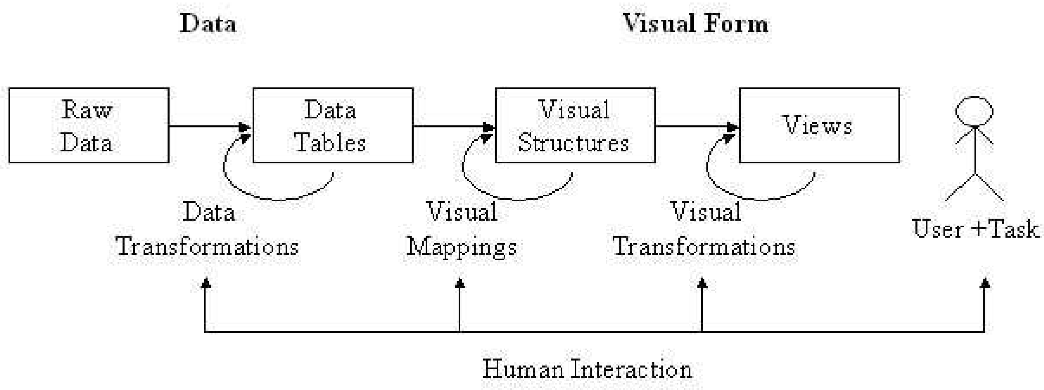


Figure 4.1: Diagram of data mappings for presentation in visual form.

From the diagram we can see that data changes from the raw format to the human suitable format through data transformations. The arrows can indicate multiple transformations. These data transformations are of the following types:

- Data Transformation: transforms, for example, Raw Data (data in idiosyncratic format) into Data Tables (relational format of data extended to include metadata);

- Visual Mappings: transforms, for instance, Data Tables into Visual Structures (structures that combine spatial substrates, marks and graphical properties); and

- View Transformations: transforms Visual Structures into Views by specifying graphical parameters such as position, scaling and clipping.

Human interaction controls parameters of these transformations, such as data ranges, nature of transformation, etc. The core of the reference model is the mapping of a Data Table, that is based on mathematical relation, to a Visual Structure, that is based on graphical properties effectively processed by human vision. An example of it is textual Raw Data that can be transformed to indexed strings or arrays, and later to document vectors and normalised vectors in a space with dimensionality as large as the number of words. Document vectors can then be reduced by multidimensional scaling to create Data Tables of x, y, z coordinates that could be displayed. Data Tables are based on mathematical relations. Relations are more structured than raw data and consequently easier to map to visual forms.

4.1.4 Classifications in Information Visualisation

Many classifications regarding information visualisation exist in the literature, regarding different aspects and perspectives. In this section we are going to cite some of them.

The work in [Shneiderman 2004] classifies data types of information visualisation in the following categories:

- 1-Dimensional;

- 2-Dimensional;

- 3-Dimensional;

- Multi-Dimensional (more then 3 dimensions);

- Temporal;

- Tree; and

- Network Data.

The work in [Lohse et al., 1994] proposes a structural classification of visual representations. It makes classification of visual representations into hierarchically structured categories. This classification is divided in six groups:

- Graphs;
- Tables;
- Maps;
- Diagrams;
- Networks; and
- Icons.

Another classification of visualisation types is proposed in [Burkhard 2004] from a perspective of architects. The visualisation types described there are:

- Sketch;
- Diagram;
- Image;
- Object; and
- Interactive Visualisation.

These existing classifications are complementary to each other and relevant in many aspects, for instance, they help understanding of how different types of visualisation can communicate knowledge, they help to identify research needs, and in addition they offer design guidance through the development of prototypes for each category.

4.2 Related Aspects

4.2.1 Users and User Interaction

There are several forms of user interaction in information visualisation, which ranges from the most basic ones to the more sophisticated. According to [Shneiderman 2004] the user can perform the following seven tasks in information visualisation:

- Overview of the data set;
- Zoom in on items;
- Filter out items;
- Details-On-Demand to select items and get details;
- Relate to view relationships among items;
- History to keep history of actions to support undo, replay, etc.; and
- Extract to allow extraction of sub-sets and of query parameters.

There are in addition other tasks that can be considered as a special form of manipulation, such as Direct Manipulation or Dynamic Queries [Shneiderman et al., 1992a], [Shneiderman 1994].

Users also benefit from works in user interaction, where Human Computer Interaction (HCI) concepts have great importance. For instance, direct-manipulation interfaces influence on creating controls as part of a presentation. In addition, interface objects are being used and proposed for interactive objects [Robertson et al., 1993], manipulation handles [Chuah et al., 1995] and interactive controls [Zhou and Houck, 2002].

The approach of User Interface Management Systems (UIMS) [Myers, 1999] provides systematic means of defining interaction controls from a syntactic/operational design focus. These types of system are designed to separate business logic from Graphical User Interface (GUI) code in the software design. UIMS are generally based on N-tier architectures and libraries and systems used as graphical tools.

Another interesting approach is *Graphical Encoding* for Information Visualisation [Matkovic et al., 2002]. It provides scientific guidance for use of graphical encoding to convey information in an information visualisation display. Sometimes inconclusive and conflicting viewpoints occur. For the graphical encoding there are visual display elements such as: icon color, shape, size, position, etc. This study suggests that the nature of the users perceptual task is more indicative of the effectiveness of a graphical encoding than the type of data represented.

In addition, in the advance design of interfaces (visual structures) for Information Visualisation, efforts are being made on supporting the systematic design of advanced user interfaces. For instance, in [Derthick and Roth, 2001] it is proposed a method to automatically generate customised interfaces.

4.2.2 Multi-Modal Visualisation

Some environments and scenarios need different modalities to support presentation. Multi-modal visualisation: (1) can be more understandable since the complementary modalities reinforces the information; (2) gives aid in a sense that the diverse modalities may enable information to be perceived in situations where visual display devices cannot be used (due to a small screen or when the user is performing remote operation); and (3) considers the aspect that the user may more easily perceive the information through one sense as opposed to another.

Recent technological advances nowadays permit users to perceive information in very distinct ways. For instance, by sound (sonification), or by means of tactile, kinesthetic or force-feedback channels. It is also possible to utilize other senses such as smell (olfaction) or taste.

An interesting work in the area of multi-modal visualisation is the Resource-Adaptive Mobile Navigation System [Baus et al., 2002], a mobile pedestrian navigation system. The adaptation of a multi-modal way description takes into account: (1) user resources, such as time pressure, working memory, familiarity, speed; (2) technical resources, for example, display size, resolution, amount of colours; and (3) quality of sensors for positioning, for instance position, orientation and speed. Regarding the technical aspects of the hybrid location sensitivity, it is based on GPS satellites (active sensing), where the mobile device detects the actual location; and on infrared (passive sensing) that presents information received from senders. For the visualisation of information, the system interface includes the presentation of graphs for route description, with possible interactions; and the adaptation of the graphical output (according to users moving speed for example, or output media).

4.2.3 Personalisation

Another relevant aspect of user interfaces for Information Visualisation is Personalisation. According to [Weld et al., 2003] an initial presumption of automatic Personalisation is that it can affect positively user productivity. Improvements can be achieved by:

- Customisation: changes guided by explicit user request; and
- Adaptation: changes based on implicit user behavior.

To provide Personalisation, projects consider different approaches, such as: user guidance by version-space algebra and model-based user interface design (declarative models).

However, some trade-off aspects are to be taken into consideration when using Personalisation in interfaces for Information Visualisation. One is the imperfection of adaptive mechanisms. To surpass some of the problems related to it, decision theory is used as a framework to analyse the cost to users caused by errors. In addition, interfaces mechanisms (e.g. timeouts) can minimise the cost of errors and improve adaptation.

4.2.4 Application Areas

Information Visualisation is applied in many areas, such as:

- Biology;
- Medicine;
- Monitoring (Process Visualisation), etc.

Approaches on how to apply Information Visualisation techniques to Process Visualisation (Monitoring) [Matkovic et al., 2002] include:

- History encoding: display values of near past and current present;
- Multi-instruments: simultaneously display several data-sources that make comparison easier;
- Levels-of-detail: uses instruments of different sizes to represent the same data (depends on screen area and amount of information). Also, techniques such as 3D anchoring, collision avoidance, focus+context rendering are used.

In Biology, Information Visualisation techniques has been used, for example, for visualising biosequence via *Texture Mapping* [Thiagarajan and Gao, 2002]. Visual data mining and the process of patterns discovery in protein (DNA) has been the predominant technique used. The visualisation approach uses: (1) texture mapping (for rendering the large set text data) and (2) blending techniques (for blending purposes), to perform visual data mining on text data. This visual approach investigates the possibilities of representing text data in 3D and provides new possibilities of representing more dimensions of information in text data visualisation and analysis. This approach contributes to derive a generic framework to visualise text in biosequence data.

4.3 Techniques for Information Visualisation

4.3.1 Information Visualisation of Hierarchies

Many approaches and techniques are proposed in the literature of Information Visualisation for dealing with hierarchies. Examples of these are:

- InterRing;
- Space-Optimised Tree Visualisation; and
- Beamtrees.

InterRing [Yang et al., 2002] is an interactive tool for visualisation of hierarchical structures. It permits visually navigating and manipulating of hierarchical structures. Some of the features of this approach are:

- Radial Space Filling (RSF): technique for hierarchy visualisation;
- Support for interactive operations on hierarchical structures (selection and navigation);
- Multi-focus distortions;
- Interactive hierarchy configuration; and
- Semi-automatic and manual selection.

One advantage of this method over other techniques, such as traditional node-link diagrams and tree maps, is the efficient use of the display space while effectively conveying the hierarchical structure. As a disadvantage, it is questioned in the literature whether it is intuitive or not.

Another approach for visualising hierarchies is Space-Optimised Tree Visualisation [Nguyen and Huang, 2002]. It consists of a method for the visualisation of structured relational data, especially very large hierarchies in a 2D space. The strategy used for that includes mechanisms such as:

- Optimise the drawing of trees in a geometrical plane;
- Maximise the utilisation of display space by allowing more nodes and links to be displayed at a limited screen resolution;

- Use of enclosure to represent tree structures;
- Modified semantic zooming technique for hierarchy exploration; and
- Calculation formalism for tree geometric layout (weight calculation, wedge calculation, vertex position).

Finally, another example is Beamtrees [Ham and Wijk, 2002] that is an approach for compact visualisation of hierarchies. Beamtrees is a method for visualisation of large hierarchical data sets. It has the following components:

- Nodes: are shown as stacked circular beams;
- Hierarchical structure: size of nodes are depicted;
- Dimensions of beams: are calculated using a variation of the tree map algorithm.

The conclusions obtained with an user study is that Beamtrees can be more effective than nested treemaps and cushion treemaps for the extraction of global hierarchical information.

4.4 New Trends in Information Visualisation

4.4.1 Information Visualisation and the Semantic Web

In recent international conferences on Information Visualisation a new trend has been given increasing attention, in the integration of ontologies and information visualisation.

Several works are proposing the application of ontologies in Information Visualisation problems and their application on the Semantic Web [W3 Consortium, 2005a]. For instance, the work in [Telea et al., 2003] proposes a graph visualisation tool that allows the construction and tuning of visual exploratory scenarios for RDF (Resource Description Framework) data. In another approach, [Fluit at al., 2002] shows how visualisation of information can be based on ontological classification of that information, by a cluster map visualisation.

In general, this new trend investigates and tries to understand the nature of the Semantic Web and its relationships to Information Visualisation. It concerns amongst other things [Geroimenko and Chen, 2006]: (1) visualisation of semantic and structural information; (2) visual interfaces for retrieving, browsing and mapping semantic information; (3) semantic-oriented use of existing visualisation methods; and (4) XML-based Internet and information visualisation.

In [Geroimenko and Chen, 2006] is discussed several aspects related to the Semantic Web, XML-based Internet and Information Visualisation. For instance, applications of ontology-based information visualisation is taken into consideration. The Spectacle system and Cluster Map, for example, have the following characteristics: (1) personalised navigation; (2) support for analysis tasks with semantics; (3) user interfaces constructed for information visualisation based on the semantic web; (4) builds on lightweight ontologies to describe domains as a set of classes and their hierarchical relationships; and (5) Cluster Map, in particular, visualises the objects of selected classes from a hierarchy, organised by their classifications.

The approach proposed in this research has similarities and is inspired by a mix of concepts of these mentioned works. It is intended to be a multi-modality visualisation framework for intelligent planning systems based on ontological representation. As future work, we are seeking also the application of the ontologies and concepts in the Semantic Web.

4.4.2 Information Visualisation and Mobile Computing

New prospects for mobile computing are emerging in the post-PC era that we are witnessing. The use of mobile devices is becoming increasingly more frequent. Mobile devices (such as pocket computers, wireless handheld devices, mobile phones, etc.) are being used more often as personal and business tools. This means that new services aimed at such devices need to be developed and improved, heading to the construction of a new mobile world. Although very limited in resources, these devices now have the capacity to run more advanced applications.

Consequently, opportunities have emerged to develop applications using several existing technologies in more diversified areas, such as Information Visualisation and Artificial Intelligence. Modalities of applications and services that have been developed aimed at desktop (fixed) platforms are now striving to meet the challenges presented by developing systems for mobile platforms. In addition to the usual difficulties of developing new systems with new technologies, in such cases there is also the aspect of dealing with a very limited platform in terms of resources. Limitations exist in all senses: processing power, memory, screen space, connection bandwidth, etc.

Recent and continuing advances in wireless networking and the fast progress of general APIs, such as J2ME [Sun Microsystems, 2003], make feasible the development of such new applications, by overcoming some of these obstacles.

J2ME, the Java Sun platform aimed at mobile limited platforms, is an open, portable (operating system and hardware platform independent) and an object-oriented API that helps the development of applications which require more advanced services, such as, agent reasoning, deduction, or other intelligent behavior. Although logic languages (such as Prolog), or languages connected with artificial intelligence research (for instance Lisp), may better match artificial intelligence paradigms, these languages are not very flexible when developing systems. For instance, systems that require graphical components for the development of user interfaces and information visualisation structures, providing particular challenges. Developing in Java APIs eases the design, integration and delivery of such systems.

Another relevant technology for the development of mobile applications is XML and its related technologies that provide data portability. The extension of the current web into the Semantic Web, based on these technologies, will permit programs to manipulate data meaningfully and automatically. The ability to manipulate the web content also increases the opportunities for new applications.

In this context, a few approaches have been proposed for the development of more advanced mobile applications, which, among other things, provide elaborated visualisation of information.

The *Resource-Adaptive Mobile Navigation System* [Baus et al., 2002] is a mobile pedestrian navigation system. It is based on location sensitivity, and for that the system considers two modalities:

- Active (GPS satellites): the mobile device detects the actual location;
- Passive (Infrared): the mobile device presents information received from senders.

Current positioning systems have been using the following technologies:

- Indoor Systems: Infra-Red and Bluetooth Radio. Examples of systems using Infra-Red are [Long et al., 1996] and [Encarnacao and Kirste, 2000]. Examples of systems using Bluetooth Radio are [Cheverst et al., 2000], [Not et al., 1998].

- Outdoor Systems: GPS (Global Positioning System), GSM (Global System for Mobile Communication) and cell based UMTS (Universal Mobile Telecommunications Systems). Examples of systems that use GPS are [Long et al., 1996], [Malaka and Zipf, 2000].

The systems also consider adaptation in a multi-modal fashion that takes into account: (1) user resources (time pressure, working memory, familiarity, speed); (2) technical resources (display size, resolution, amount of colours); and (3) quality of sensors for positioning (position, orientation, speed).

In [Elting et al., 2002] an empirical study is made of device dependent modality selection. The aim of this study is to investigate the effects of multi-modality use for information visualisation in different devices (where the best modality might depend on the device). The experiment consists of:

- Devices used: desktop PCs, TV set with remote control, PDAs;

- Modalities used: written text only (T), written text with the same text presented as spoken text (TS), written text with picture (TP), written text with spoken text and picture (TSP), spoken text with picture (SP).

The users were questioned about how much information was learnt. According to the study the results are: (1) Text/Picture/Speech are the most appealing modalities for users; (2) Picture/Speech are the most effective; (3) combination of modalities on PDAs are not useful due to the cognitive loading.

An important module of the experiment is the *Presentation Planner* that adapts the presentation to the cognitive requirements of the device used, consequently avoiding cognitive loading.

4.4.3 Information Visualisation and Ubiquitous Computing

Vanguard projects in Ubiquitous Computing are addressing issues in Information Visualisation. In this subsection we discuss some projects with this focus. The systems use different modalities for user interfaces in information visualisation tasks.

The EXACT [Yates et al., 2003] system is based on a natural language interface for household appliances. The motivation for this project is that household appliances are growing in complexity and sophistication, thus becoming harder to use. This is enhanced by the fact that the appliances have tiny display screens and limited keyboards. The project proposition is to offer a natural language interface for household appliances. The approach used is based on research in planning and natural language interfaces to databases. As such, it reduces the problem to a database problem. The system executes a mapping from an English request to a database SQL query, and afterwards maps to a goal in PDDL, the Planning Domain Definition Language [McDermott et al., 1998], that is subsequently sent to a planner, that finally maps this to a sequence of appliance commands.

In addition, groups, such as, the Wearable Group at Carnegie Mellon University (CMU), Vision Group at Microsoft Research, Oxygen Project and Vision Interface Group at Massachusetts Institute of Technology (MIT) are also investigating new ways for Information Visualisation in Ubiquitous Computing.

The Wearable Group [The Wearable Group, 2003] at CMU has an interdisciplinary team performing research into the architectural and interface requirements of wearable systems. They consider multi-modal interfaces for Information Visualisation, for instance, audio and tactile interfaces.

The Vision Group [Vision Group , 2003] at Microsoft Research develops the Easy Living Project in Ubiquitous Computing. They are developing a prototype architecture and technologies for building intelligent environments, using: (1) technologies of Computer Vision for person-tracking and visual user interaction; (2) fine-grained events and adaptation of the user interface; and (3) device-independent communication and data protocols.

The Oxygen group [Oxygen Project , 2003] at MIT uses pervasive human-centered computing technologies to directly address human needs. Speech and vision technologies are used as communication interfaces with machines, devices, actuators, and sensors. They work on perceptual interfaces for information visualisation, multimodal systems, and multilingual systems.

The Vision Interface Group [Vision Interface, 2003] at MIT investigates ways to make computers more natural and easy to use, using machine perception techniques, and vision based perceptual interfaces.

Chapter 5

Information Visualisation in Intelligent Planning Systems

5.1 Introduction

Information visualisation is an important area of intelligent planning systems since it can provide, among other things, ways to improve the interaction between users and advanced planning services and resources. However such an area is still not very well explored because the principal efforts in the AI planning field are mainly focused on problems of planning efficiency regarding plan generation and representation. Examples in this directions involve proposals for more efficient search algorithms or shortest plans such as the works presented in [Long and Fox, 2003], [Brafman and Hoffmann, 2004] and [Zhou and Hansen, 2005].

Due to these classic research directions, there is a lack of research that addresses the problem of information visualisation. This problem is even more important for collaborative planning systems. In those, the different participants will have different backgrounds, play different roles, and have different capabilities and responsibilities, etc. That makes more complex the task to adapt information visualisation to the agents requirements. Therefore, advances in AI planning technology evidence a need for more sophisticated approaches to planning information visualisation that will mirror the updated underlying technology, in contrast to planners working in an isolated way in the past.

This chapter will investigate the mechanisms and features that have been used for traditional AI planners, so that we can answer the following questions:

- Which are the common methods of information visualisation in planning systems?

- Why are such methods being used? Are there any obvious advantages in their use?
- Are the methods directly related and dependent to the planning approach?
- Is there a strong relation between the kind of domain and the visualisation? Or are such concepts independent of each other?
- Which are the opportunities to advance the state of the art in information visualisation in such systems?
- Which are the common technical details used by such systems (information filtering, colour as form of differentiation, etc.)?

Note that such an investigation is very important because we intend to define an automatic reasoner that tries to match the most appropriate style of visualisation to the kind of information generated by the systems. Thus, several clues can be raised from this study.

Another relevant objective of this investigation is to raise the additional issues to visualisation associated with the use of planning systems in collaborative environments. Certainly such environments require a new set of information, which is not common in traditional planning systems. Thus, the study of collaborative planning systems allows us to highlight the additional requirements, showing how they are being faced by current systems. The same questions, discussed in relation to traditional systems, can be used to guide our study of the more general approach here.

The analysis will be made by considering the different phases of the planning processes, such as plan generation, collaboration, execution, re-planning, etc. Hence, this study will be able to show how information has been manipulated and presented in each step of the process.

The next sections are organised as follows: section 5.2 presents a chronological view of important planning systems, so that we can define if there is some kind of visualisation trend in this area. Section 5.3 gives an overview of such systems, taking information visualisation as the main focus. Section 5.4 is a deeper investigation with a categorisation analysis according to distinct perspectives of comparisons. At the end of Part II, in Chapter 6, we summarise the problems, gaps and research opportunities in the area.

5.2 A Chronological View

It is important to note that the technological restrictions of a specific period is an important factor to be considered when we are discussing visualisation methods in any kind of system. For example, visualisation methods developed for more than 20 years old planners (sometimes graphics presented as a set of ASCII characters) certainly did not have the same resources as current planners (usually GUI-based interfaces). Thus it is not so effective to trace a comparative discussion between such planners because the principal factor of differentiation is essentially the technological restrictions rather than parameters such as domain, planning approach or user role.

However, there may be interesting information that we can extract from an analysis of the historical development of planners, and we can look to see if there is some kind of trend leading to the development of visualisation mechanisms. Based on this idea, we have used time as the ordering factor to discuss the planners in the next section. This chronological view[1] is illustrated in the follow (Figure 5.1).

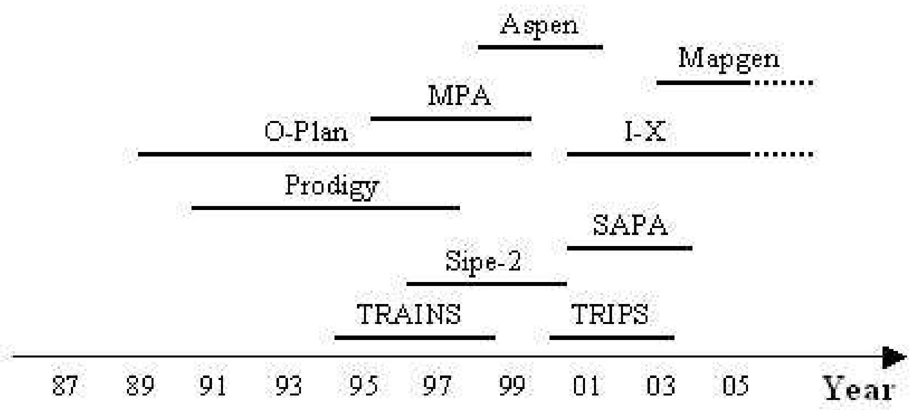


Figure 5.1: Chronological view of some important planners.

Independent of technological restrictions, the existing planning systems use different approaches for information visualisation. Some of the AI planners give emphasis to the search algorithms for efficiency, rather than exploring for example, user inter-

[1] Intervals in this graphic represent the principal period of development of the systems.

action thorough information visualisation. Other systems only opt for one category of information visualisation, when this solution is sometimes not adequate to every situation. Furthermore, there are many other important aspects to be explored including: collaboration, different types of users and their roles in the planning process, devices differences, and type of information to be manipulated.

5.3 Systems Overview

This section presents an overview of existing AI planning systems in respect of information visualisation. The relevant AI planners selected for this study were (in chronological order): O-Plan (and its successor I-X), PRODIGY, TRAINS (and its successor TRIPS), SIPE-2, MPA, PASSAT, ASPEN, MAPGEN, and Sapa. For this study we first introduce the systems, discussing their approaches for plan generation and representation. After that we focus on information about their information visualisation methods.

5.3.1 O-Plan and I-X

O-Plan [Currie and Tate, 1991] is a knowledge-based and hierarchical task network planner. It provides an environment for specification, generation and execution of activity plans, and also uses interaction with generated plans. O-Plan is based on the earlier Nonlin [Tate, 1977] planning system developed at The University of Edinburgh.

O-Plan is intended to be a domain-independent planner, where detailed knowledge of the domain can be used. O-Plan uses the <I-N-OVA> (Issues - Nodes - Orderings / Variables / Auxiliary) constraint model to represent plans and processes. Later on, some of the O-Plan concepts, such as the plan representation model, evolved in its successor, the **I-X** system which uses the <I-N-C-A> (Issues - Nodes - Constraints - Annotations) [Tate, 2001] model (Figure 5.2). The O-Plan hierarchical planning system produces plans as partial orders on activities, and additionally it has an agenda-based control architecture to control problem solving cycles during plan generation.

O-Plan provides three types of interfaces: a GUI interface, a web interface, and a limited media interface. The GUI interface [Tate and Drabble, 1995] considers roles played by users in the planning process, providing different views of plans. The roles available in O-Plan include: task assigner, planning specialist and operational execution staff. The O-Plan GUI was built based on a Computer Aided Design (CAD)

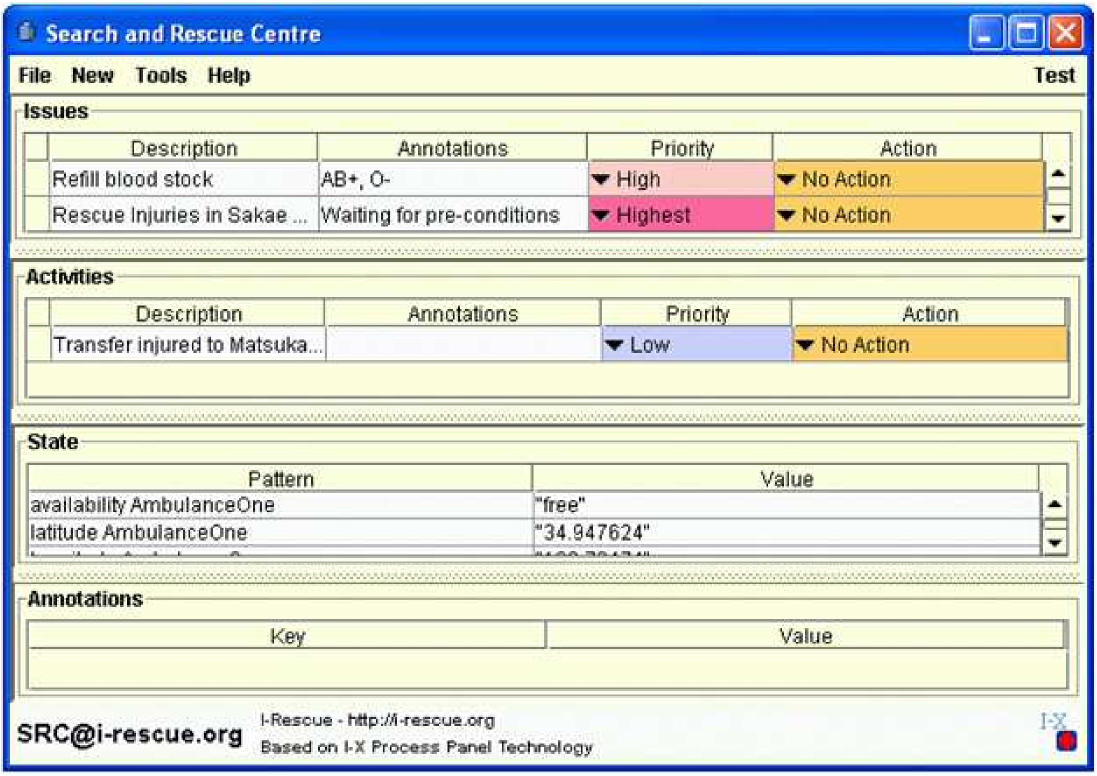


Figure 5.2: I-X Process Panel.

package - AutoCAD. The types of planning information available are related to: specification, generation, and execution of activity plans, also allowing interaction. This set of information is presented to the user in two views: Plan Views and World Views. The Plan Views is the interface used to show charts, structure diagrams, etc. of the plan, and the World Views permit visualisation and presentation of simulations and animations.

The O-Plan web interface (Figure 5.3) [Tate et al., 1998] is part of a web-based demonstration. It permits a task assigner user and a planner user to interact with the O-Plan planning system (where multiple users and systems in different roles work together in a mixed-initiative fashion) to explore different options for constructing multiple Courses-of-Action (COAs), and displaying these in a COA evaluation matrix.

This interface provides a table where the columns show the options for each COA and rows show the process steps involved in generating the plans and, in addition, a set of evaluations of the plan options. The domains used in the demonstration are logistics and crisis operations domains: Pacifica Disaster Relief, Pacifica Non-combatant Evacuation Operations (NEO), US Army Small Unit Operations Military Operations

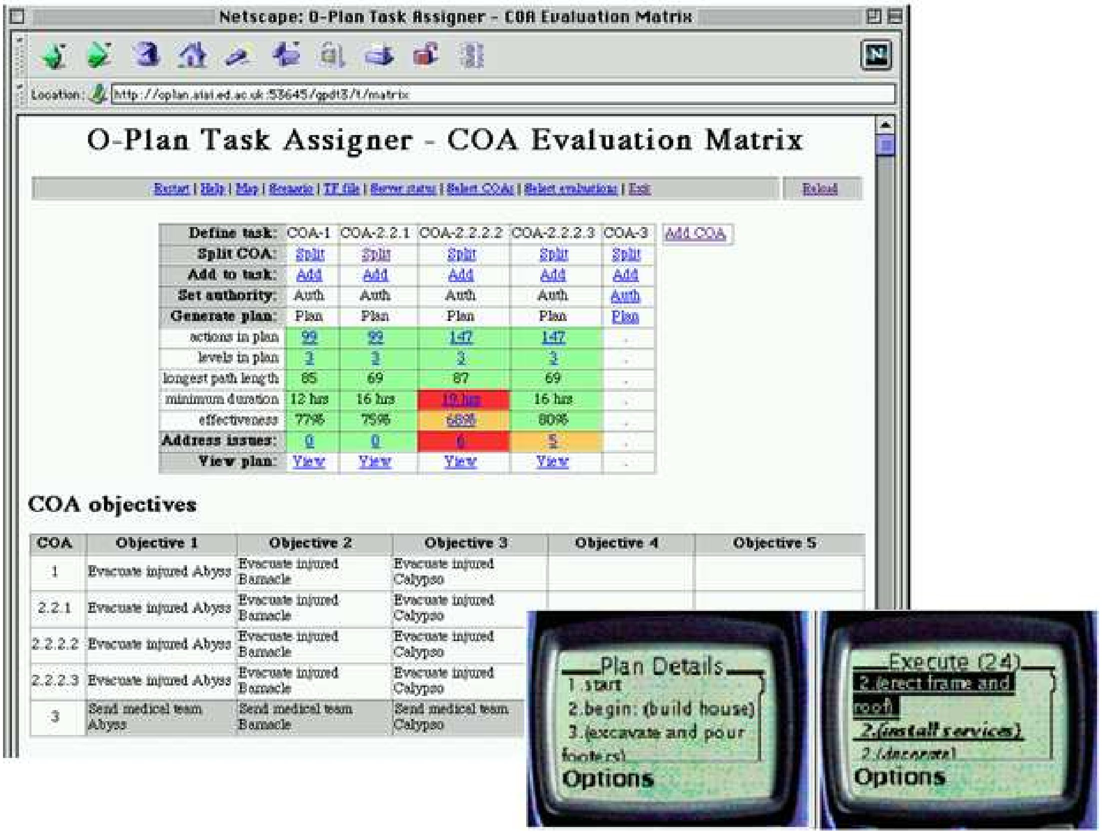


Figure 5.3: O-Plan web and O-Plan limited media interfaces.

in Urban Terrain (SUO MOUT), and Generic COA Evaluation Matrix. Via the web interface it is possible to run O-Plan remotely over the Internet, have interaction between different users in different locations, and produce plans for tasks in these different domains. Additionally, it has facilities for interaction with the system in a mixed-initiative style during plan generation, and also for simulation of plan execution and plan repair [Tate et al., 2000], [Tate, 2000]. The web interface is defined in more detail in [Tate and Dalton, 2003].

When using the web interface, the user is initially given a blank COA evaluation matrix, which is populated by the user and the planner during the demonstration. One user assumes the role of 'Task Assigner', whose functions are to: define the initial assumptions and tasking level requirements for a COA, and selecting elements of evaluation to include in the matrix. Any COA can be divided into two or more alternative options by the 'Task Assigner', and also additional constraints can be added. A second user assumes the role of 'planner' and can then refine the plans and generate more options. Some of those can be passed back to the Task Assigner user. The results (plans and others) are available via web links.

Finally, the O-Plan limited media interface [Nixon et al., 2000], [Tate et al., 2003] consists of a mobile telephone interface (Figure 5.3) called WOPlan (Wireless O-Plan). This interface was developed as a Java Servlet application, which communicates with the O-Plan system. This interface is aimed at WML (Wireless Mark-up Language) mobile telephones. In this approach a simple planning execution facility is included, not present in the standard O-Plan GUI. To execute a plan the user is presented with a depth-first ordered list of the activities in a hierarchical plan that have the status of being executable now, given what has been completed so far in the execution process. Through calls to the servlet, the current execution state of the plan is updated. As this kind of device has very limited screen space, information is presented with the reduction of any graphical interface in order to maximise the usability of the limited media interface.

5.3.2 PRODIGY

PRODIGY [Veloso at al., 1995] is a general-purpose planner that has learning modules to refine the planning domain knowledge and the control knowledge with the objective of guiding the search process effectively. In its first design, the project focus was on how to integrate learning and planning. The main characteristics of the system (in the first version) were that: the planner assumed a linear sub-goal decomposition (i.e., no interleaving of sub-plans), the learning technique used was explanation-based learning of control knowledge to guide the search process, and the architecture included empirical analysis of the effect of learning control knowledge on the planner's performance. In the next phase PRODIGY investigated alternative learning techniques to address more complex domains and problems. The planning algorithm went from simple linear and incomplete (Prodigy 2.0) to non-linear and complete (Prodigy 4.0). The architecture was developed with several learning methods that improved the performance of the core planner.

Related to its plan representation, PRODIGY's language for describing operators is based on the STRIPS [Fikes and Nilsson, 1971] domain language, extended to express disjunctive and negated preconditions, universal and existential quantification, and conditional effects. PRODIGY uses both partial-ordered and total-ordered plans. In the system, an incomplete plan consists of two parts: the head-plan and tail-plan. The tail-plan is built by a partial-order backward-chaining algorithm, while the head-plan is a valid total order plan. Regarding the planning algorithm, in PRODIGY the planning domain is specified as a set of operators, where each operator corresponds to a generalised atomic planning action, described in terms of its effects and the necessary conditions that enable the application of the operator. A planning problem in a domain is represented as an initial configuration of the world and a goal statement to be achieved. In this way, a planning domain is defined as a set of typed objects: classes used in a domain, library of operators, and inference rules that act on these objects. Inference rules have the same syntax as operators, and each operator is defined by its preconditions and effects.

Regarding its user interface, PRODIGY has a graphical user interface that was built by integrating the planner with off-the-shelf software components. The interface permits the creation and use of plan domains, and its design intends to be modular and extensible. Communication between the two processes (planner and interface module) is implemented with sockets and agreed messages. The PRODIGY user interface permits a certain level of interaction with the planner, for example, the user can follow an animation of the algorithm, interrupt the process to analyse the details, and change to different planning search strategies. Figure 5.4 illustrates the PRODIGY user interface.

The main functions available in the PRODIGY user interface are: (1) a visual animation of the planning procedure and visual representation of the output, (2) help for the process of creating and debugging domains, and (3) provision of an uniform access to the modules built on top of PRODIGY. Extensions of the user interface can permit planning by analogical reasoning and probabilistic planning.

It is possible to create domains in PRODIGY in three ways: (1) create the Lisp structure directly, (2) using the APPRENTICE system that produces the domain from a graphical specification, and (3) via a form-based tool called Domain Builder that allows interactive domain development within the planning system.

The user interface is implemented in the Tcl/Tk scripting language, which includes a set of widgets, and uses a freely available processor for drawing directed graphs. The user interface has the advantage of being flexible as it can ideally be integrated with variants of the system without the need to make changes in the planners code. This is due to the interface making very few assumptions about the planner implementation, but at the same time the interface is tightly integrated with the planner which permits planning related information to be shown graphically in the interface. Another advantage is that the use of off-the-shelf components enables a quick development.

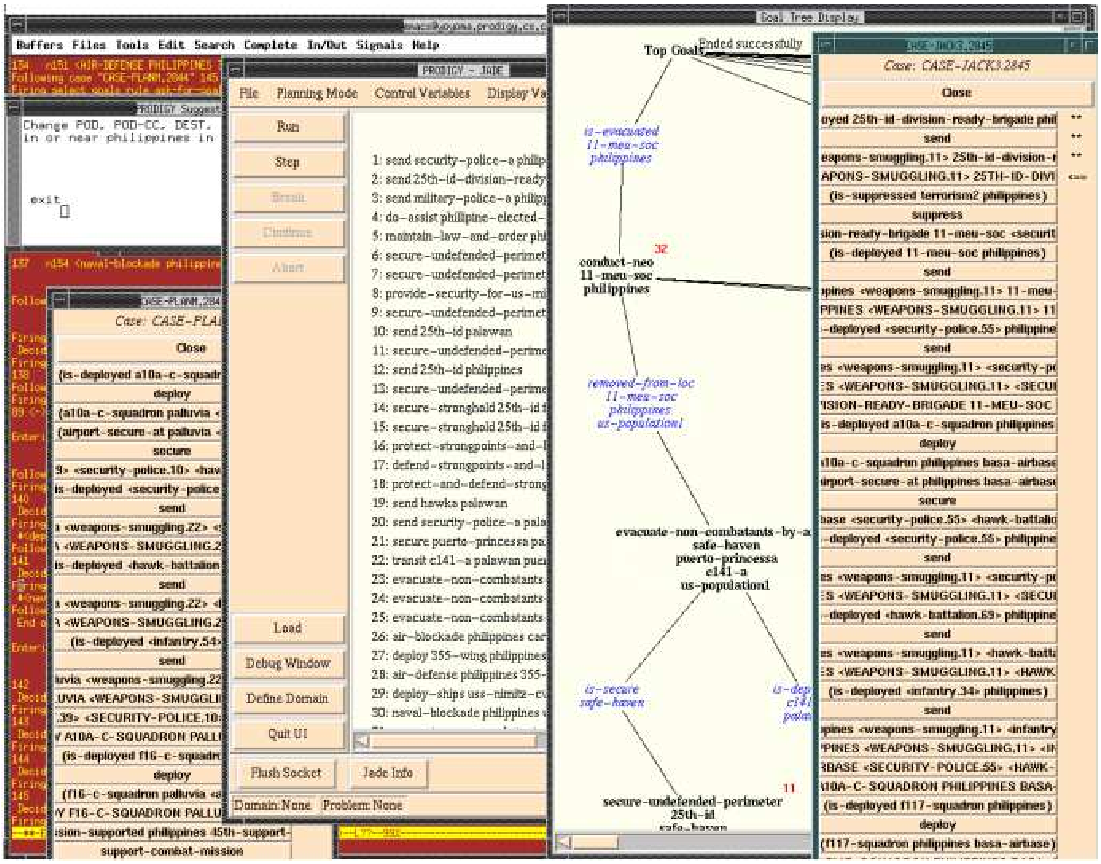


Figure 5.4: [Veloso et al., 1998] - PRODIGY user interface.

5.3.3 TRAINS and TRIPS

A more distinct approach for visualisation and user interfacing is used in the **TRAINS** system, and its successor, the **TRIPS** project. The approach of these systems is more distinctive in a sense that it is not predominantly based on GUIs, but on natural language processing. In [Allen et al., 2001a] and [Allen et al, 2001b] are discussed the natural language user interface approach of the TRAINS and TRIPS systems. The architecture of these systems is based on an integrated set of technologies and tools to assist intelligent problem solving. More specifically, TRAINS and TRIPS are systems that support spoken and written language dialogue to collaboratively solve planning problems. Figure 5.5 illustrates the TRIPS system user interface.

Related to plan representation, a shared representation of plans is used among the components of these systems. The application domains are characterised as realistic logistic domains of small complexity. Initially, the domains were based on routing and scheduling of trains. Later, territorial evacuation scenarios, like the *Pacifica Domain*

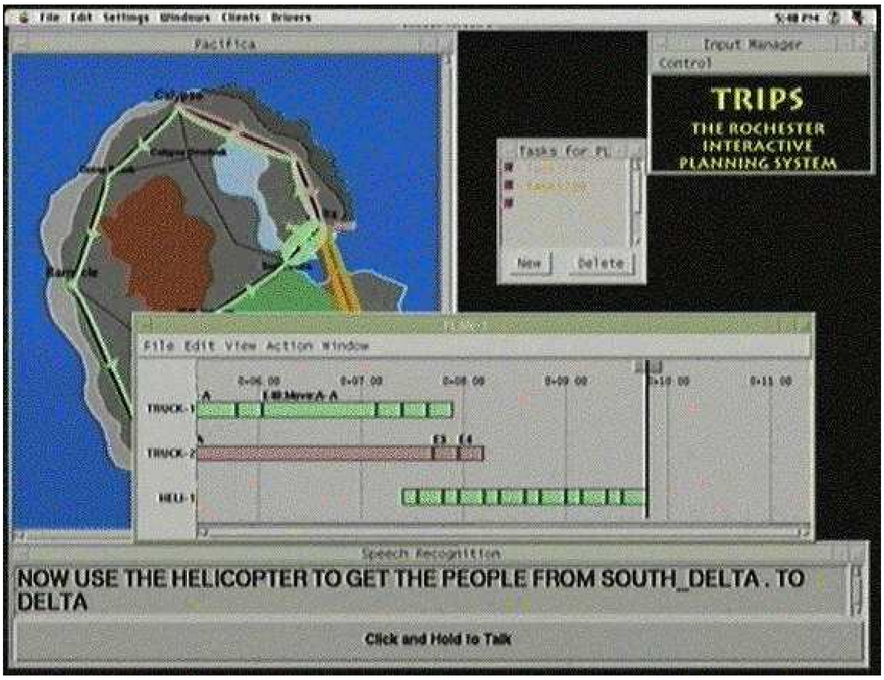


Figure 5.5: [Ferguson, 2000] - TRIPS user interface in the Pacifica domain.

[Reece et al., 1993] was considered.

The user interface approach aim is to apply natural language dialogue to solve planning problems in a collaborative way. Interactions can be done either by spoken or typed English and involves defining and discussing tasks, exploring ways to perform tasks, and collaborating to execute tasks. Using natural language processing, interactions are also contextually interpreted. In addition, to illustrate what is happening in the process, map based visualisation is jointly exploited in which maps are used and updated according to the actions taken.

Interfacing collaborative systems and their users with natural language techniques is an alternative and valuable modality. In many situations it could be the most appropriate approach, for instance, in situations where users are using their hands and/or eyes in parallel activities, so that interacting without hands/eyes could help. Also, it is suitable for devices with limited screen space, since information delivered by voice can free space on the screen. However, some researches claim that GUIs are not to be entirely substituted by speech recognition, as for example in [Shneiderman, 2000], but instead, these modalities should complement one another.

5.3.4 SIPE-2

SIPE-2 (System for Interactive Planning and Execution) [Wilkins, 1999] is an interactive planner system that permits human input during planning. SIPE-2 is based on partial-order AI planning, and supports planning at multiple levels of abstraction. SIPE-2 uses the Act Formalism [Wilkins and Myers, 1995] (actually MPA system, to be described later, uses SIPE-2 as the planner). SIPE-2 utilizes knowledge encoded in this formalism and heuristics for reducing the computational complexity of the problem, to generate plans for achieving the goals. Given an initial situation, the system either automatically or under interactive control combines operators to generate plans for achieving the goals. The generated plans include information that permits its modification during plan execution if the system has any unexpected occurrences. In addition, the SIPE-2 framework allows reasoning about resources, the posting and use of constraints on plan variables, and the description of a deductive causal theory to represent and reason about the effects of actions in different world states.

Related to its visualisation approach, SIPE-2 has a graphical user interface built also (as MPA is) on Grasper-CL [Karp et al., 1994], a system that allows viewing and manipulating graph-structured information and building graph-based user interfaces. Its graphical resources permit: inputting domain knowledge and creating operators; following and controlling the planning process; and the graphical viewing of planning information (plans, operators, world descriptions). Some resources are also available in the SIPE-2 GUI for expert users with a strong background in planning technology, such as a Lisp listener panel. The system also has mechanisms to define layout and adjust the information to be displayed on the screen. For example, it gives the user options to choose which actions to display and what information to display for each action. Colour and shape are used to distinguish information, such as goals to be solved and actions.

SIPE-2 also has graphical tools for knowledge acquisition. The SRI Act-Editor supports graphical displaying, editing and imputing of Acts, the basic unit of representation of the Act Formalism. In addition, a SIPE-2 sort (type) hierarchy for objects can be created, viewed and edited using SRI's generic knowledge base editor, the GKB-Editor.

In the SIPE-2 GUI, users familiar with planning technology can use many resources to control the planning process and interact with the planner. For instance, the user can decide when to apply certain planning algorithms (plan critics), choose which operator to apply after the system has determined the ones applicable, inspect data structures (display an operator before choosing it), opt for planning automatically for either one abstraction level or for the rest of the plan, understand what the planner is doing by highlighting a node on the screen whenever the system is making a decision about that node. Also it is possible: (1) to highlight actions involved in resource conflicts when interactive solution of resource conflicts has been requested, (2) chose of two actions for ordering where the GUI gives a visual depiction of how the plan is flowing and (3) there is a movie-mode facility to be used during automatic planning. However, despite SIPE-2 having many visual resources in its GUI, they are not suitable for users with limited backgrounds in AI planning technology.

Figure 5.6 illustrates the SIPE-2 GUI, where on the left side is displayed the commands of the drawing menu, and on the right side is shown a graphic representation of a plan at a high level of abstraction. There is a semantic notation that says that green hexagons are goals still to be solved, and blue capsules are actions.

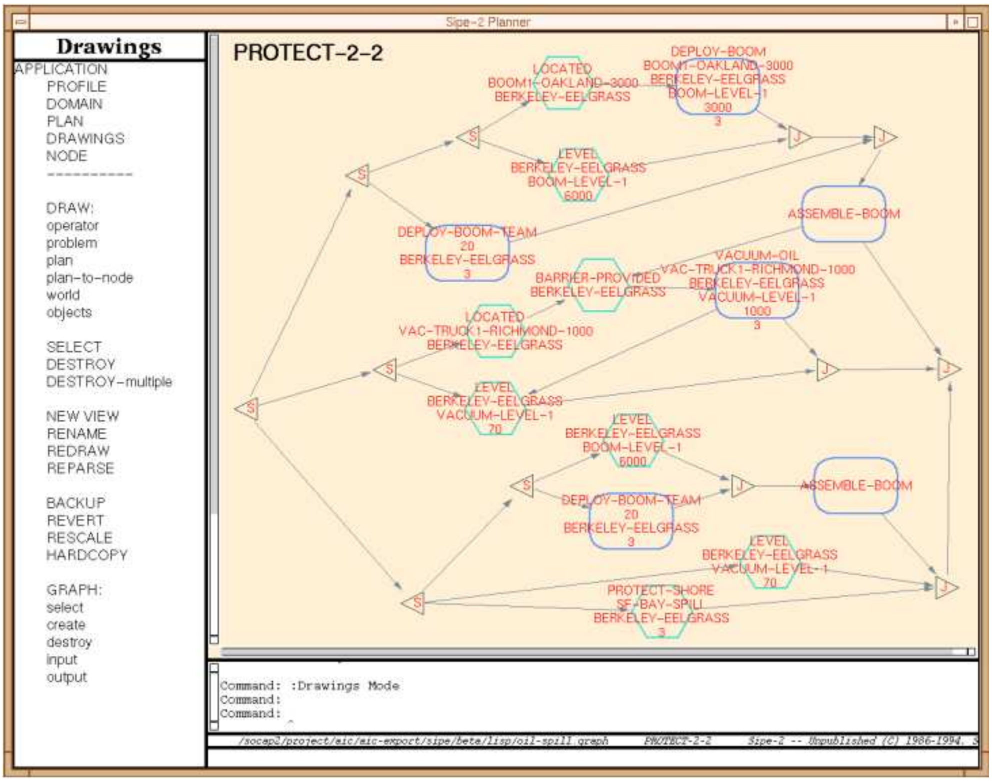


Figure 5.6: [Wilkins, 1997] - SIPE-2 GUI.

SIPE-2 also has resources to permit viewing of large plans. When expanding the plan to the lowest levels it can contain hundreds of actions, so the plan drawing doesn't fit on the screen, making it difficult to visualise. To surpass these problems, the interface provides some techniques, such as scrolling, a birds eye view that shows the plan in a low resolution window which controls the view in the high resolution, options concerning which actions to display and what information to display for each action, and also commands in the node menu that are useful when analysing large plans.

5.3.5 MPA

The multi-agent architecture for planning **MPA** (Multi-agent Planning Architecture) [Wilkins and Myers, 1998] uses an approach of an open architecture to permit integration of different technologies to solve planning problems in large-scale domains. It is designed to solve problems that require the use of combined technologies and cannot be solved by individual systems. In MPA, interface specifications are shared by agents which makes possible the integration of different technologies. The system has a centralised storage approach for plan-related information in a shared plan representation, and meta-level agents that control and customise the interactions between other agents. MPA's planning representation approach is called the Act Formalism (Wilkins and Myers, 1995), which is a language for representing knowledge about the generation and execution of plans in dynamic environments. Agents in the MPA multi-agent architecture approach share this language and interface specification to integrate different technologies in the system.

For information visualisation, MPA has the integration of agents that are responsible for the roles of (1) user interaction; (2) plan visualisation; (3) plan evaluation and simulation output visualisation. These agents are implemented making use of legacy systems: the ARPI Plan Authoring Tool (APAT) from ISX, VISAGE system from MAYA, and Air Campaign Simulator (ACS) from the University of Massachusetts. The APAT agent has the role of the user interface, advice manager and plan visualization. The VISAGE agent is also responsible for plan visualisation and, while the ACS agent provides simulation of plans, the VISAGE agent also provides plan visualisation for simulation outputs.

The visualisation approach in these legacy systems, and their respective agents (for user interface and plan information visualisation) is based on GUIs. These systems are based on different technologies: APAT is a legacy system written in Java, while, for instance, the ACS system is written in LISP. However, MPA agents integrate these different technologies in the system. In order to make possible this integration in the

MPA multi-agent planning architecture, the agents read the planning information represented in the Act Formalism and translate it to their own representation.

Figure 5.7 [Wilkins, 2000] shows VISAGE/MAYA data plots for information produced by ACS from a planning simulation on a MPA demonstration. This demonstration was entitled 'Planning and Evaluation of Multiple Alternatives using Advice, Visualisation, and Simulation'. This specific data plot shows air strikes by target status and aircraft status.

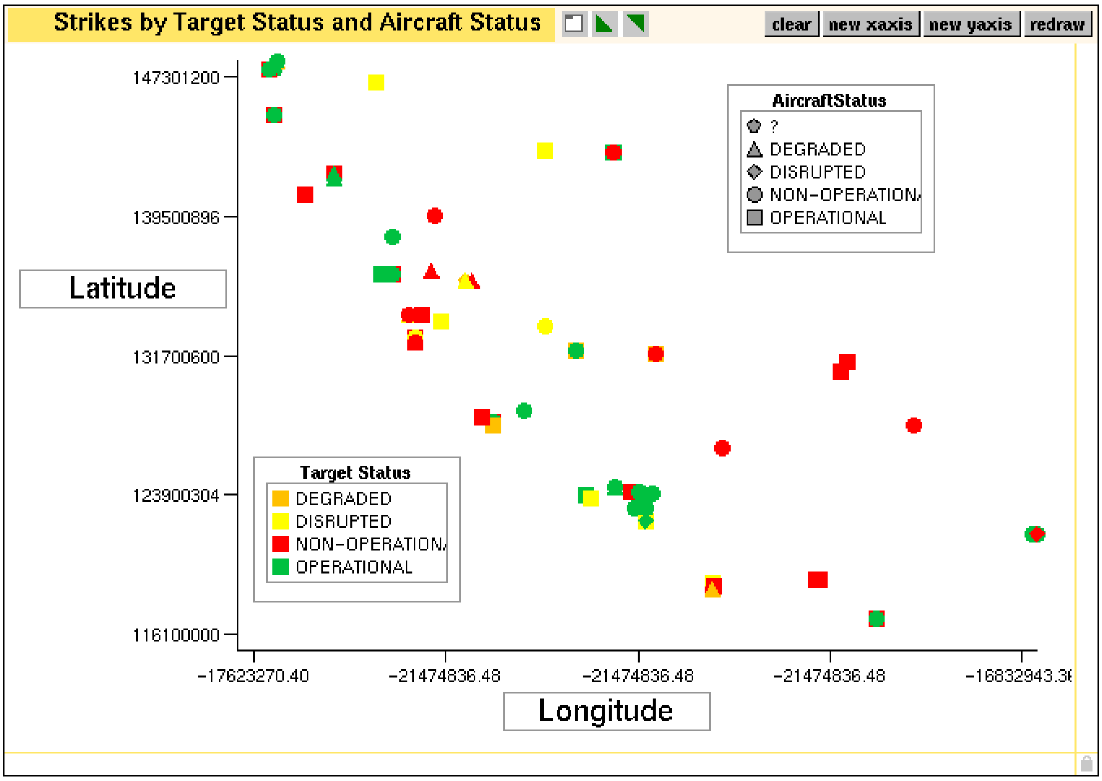


Figure 5.7: [Wilkins, 2000] - VISAGE/MAYA data plots for planning simulation information.

In addition, there is also an editing system for the Act Formalism, the Act-Editor, which is a graphical browsing and editing system for knowledge expressed in the *Act* language. Through the Act Editor it is possible to create, view and edit Acts, the basic unit of the *Act Formalism*. Each Act describes a set of actions that can be taken to achieve specified goals in certain conditions. The Act-Editor also permits browsing graphical procedures, editing procedures through direct manipulation, managing plans and operating procedures, and verifying against dictionaries of predicates and objects. User interfacing is done by a graphical display, based on Grasper-CL [Karp et al., 1994] software. Grasper-CL is a system for viewing and manipulating graph-structured information and for building graph-based user interfaces for application programs.

Considering MPA's visualisation approaches, one view is that, although MPA has not itself got a plan visualisation implementation, it can benefit from its architectural nature - multi-agent and multi technology - to integrate different services that can provide different solutions for information visualisation. For instance the ones provided by legacy systems, off-the-shelf software or even new and/or customised solutions. Additionally the sharing of the same plan representation approach - the Act Formalism - is a positive aspect of the MPA system, including for visualisation purposes. The use of a standard representation for planning related information permits it to be used as an input for information visualisation components. In this way, plan visualisation agents can use this representation of plans, operators and operating procedures as input, and integrate visualisation components and technologies to the system. Nevertheless, from the demo project GUI, showed in Figure 5.7, can be noted that this information visualisation solution is completely customised for the application context and goal.

5.3.6 PASSAT

Plan-Authoring System based on Sketches, Advice and Templates [Myers et al., 2002], or PASSAT, is a plan-authoring system that supports the user in mixed-initiative processes of planning. Plan authoring systems provide a set of plan editing and manipulation capabilities that support users in developing plans. Such systems provide new ways to structure the planning process through principled representations of plans with well-defined semantics.

PASSAT has tools for constructing plans and modules for automated and mixed-initiative planning designed to complement human skills. Using PASSAT users can construct and modify plans interactively and draw upon a library of templates to assist the plan process. Templates are a form of hierarchical task networking (HTN) [Tate, 1977] and contain parameterised standard operating procedures and cases.

The system has two principles for planning, in a combination of interactive and automated capabilities:

- **Flexible *out of the box* planning**: works as a traditional AI planning system. Offers the users a set of solutions in a form of predefined action models that underlie plan deployment. The solutions are based on templates, however it works only as a guideline for performing tasks. The user has flexibility to expand the set of solutions defined by the template (for instance the human planner can override constraints, drop tasks, or insert additional tasks). This flexibility is good for domains where correct and comprehensive templates cannot be provided.

- **Controllable user-centric automation**: automation designed to complement human skills that is invoked under user control in contexts where the human planner feels it is beneficial.

PASSAT'S approach for plan representation is based on the HTN [Tate, 1977] model, but with extension for temporal representation for tasks. Regarding mixed-initiative style for planning, PASSAT has the following main features that supports it:

- A library of predefined *templates* that encodes task networks of standard operating procedures and previous cases;
- A mixed-initiative plan *sketch* module that permits users to refine outlines for plans to complete solutions;
- An *advice* capability that permits users to specify high level guidelines for plan that the system helps to enforce; and
- A *process facilitation* mechanism that allows user to keep track and manage planning tasks and information requirements.

These mixed-initiative features appear in the PASSAT GUI. Figure 5.8 is an example of a snapshot of PASSAT GUI during a planning section. The components of the interface are as follow:

- Large left frame: contains hierarchical decomposition of current partial plan. Folder icons represent tasks that have been expanded, star icons represent tasks that can be expanded further (automatically or interactively), and documents icons are tasks that match no template;
- Upper right frame: shows the current agenda (the list of planning steps the user must perform to address outstanding issues);
- Lower right frame: displays list of information requirements (source of information that has been identified by the user, or PASSAT'S planning knowledge as relevant to various portions of the planning process);

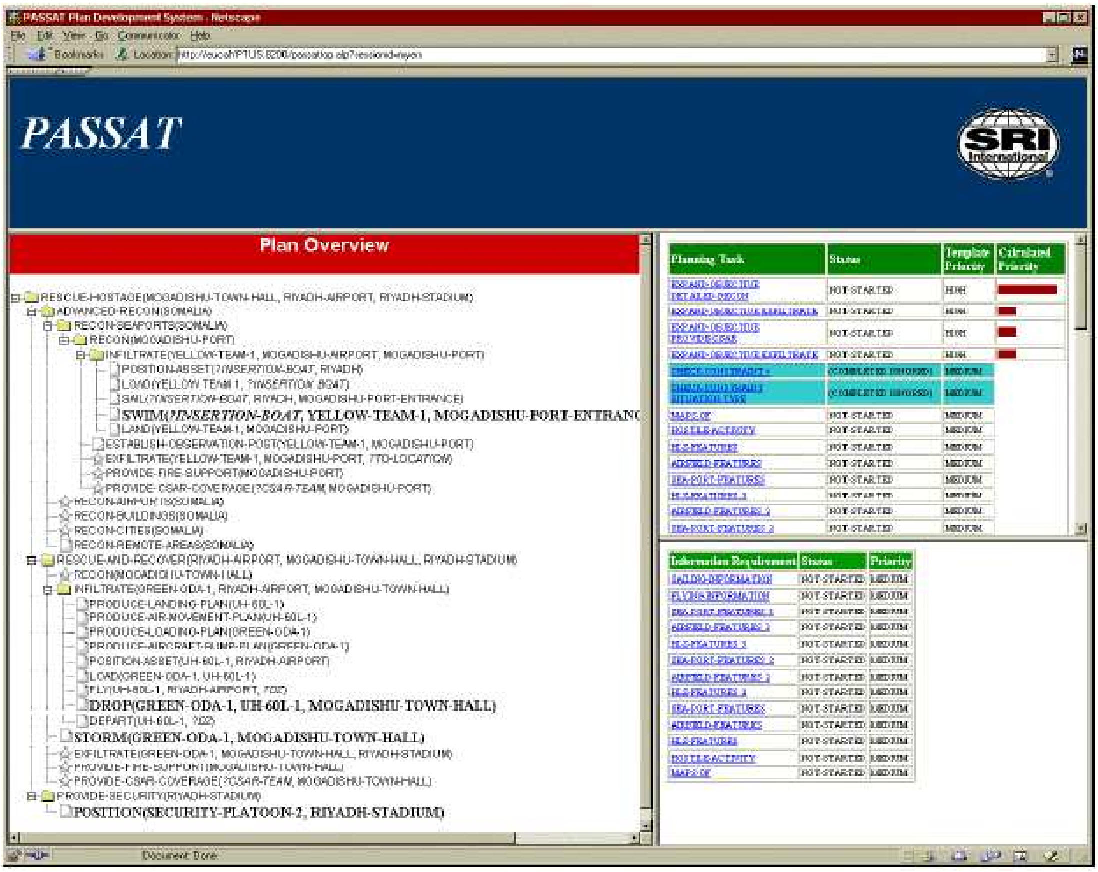


Figure 5.8: PASSAT user interface - [Myers et al., 2002].

The interactive process through the interface occurs when the human planner develops the plan by selecting a planning step from the agenda and performs that step. The planner then would be presented with several options, such as: (1) apply one of the templates that matches the task, (2) enter an expansion manually, or (3) create a sketch for achieving the task and work with PASSAT to refine the sketch. Processing a planning step like that can generate additional planning steps to be added to the agenda and also new requirements. Basically, there are two main modes of user-centric plan development:

- Interactive plan refinement: this mode involves three types of planning step that the user can interact - expand task, instantiate variable and resolve constraint;

- Plan sketching: in this mode the user can sketch an outline of a plan with the system providing assistance in expanding the sketch to a solution for a particular objective.

Figure 5.9 shows a GUI for the interactive process of sketching, where the user assists in the repair of an original sketch.

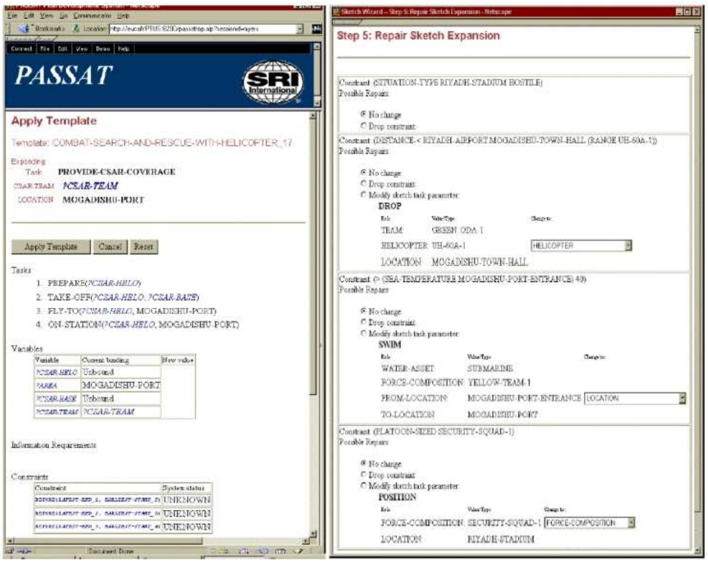


Figure 5.9: PASSAT interface for user interaction.

The window displays the available repair options for each violation that occurs, which may consist of: (1) dropping the constraint, (2) changing a parameter for a designed task, or (3) making no repair. For supporting the user in changing a task parameter, the interface provides a drop-down list of candidate values checking before for violations of the constraint in question.

5.3.7 ASPEN

ASPEN (Automated Scheduling and Planning Environment) [Chien et al., 2000] is an integrated planner and scheduler system designed for space mission operations, where planning and scheduling operations consists in generating, from a set of high level science and engineering goals, low level spacecraft commands. These low level commands include coding of spacecraft operability constraints, flight rules, spacecraft hardware models, science experiment goals, and operation procedures. ASPEN permits automation of command sequence generation and encapsulation of operation specific knowledge, which can be controlled by a small operation team.

The system provides planning and scheduling services through the following features: a constraint modelling language, a constraint management system, a set of search strategies for plan generation and repair, a language for representing plan preferences, a real-time replanning capability, a temporal reasoning system for temporal constraints, and a graphical interface for visualising plans and schedules in a mixed initiative fashion of planning.

ASPEN GUI (Graphical User Interface) allows manual generation and manipulation of activity sequences. The GUI is time oriented and composed of components that permit: (1) plan modification via pull down menus and buttons; (2) visualisation of activities as black horizontal bars; and (3) display values of resources and state variables over time as coloured blocks in the bottom part of the GUI

ASPEN has been applied in space related applications, such as:

- Distributed Self Commanding Robotic Systems: for operation of multiple spacecrafts;
- Citizen Explorer (CX1): a small earth satellite where ASPEN is used to automatically generate its command sequences.

Figure 5.10 shows snapshots of the ASPEN GUI during a demo of the Citizen Explorer project. Time is on the horizontal axis, where later times are shown on the right. The upper part of the screen shows the current activities in the mission plan, with each line beginning at the activity start time and finishing at its end time. At the bottom, the time lines represent the state and resource evolution according to how it is modelled and tracked by the planner.

In the ASPEN system the model is a description of the types of activities that can be performed on the spacecraft, together with constraints imposed by the spacecraft on those activities. Constraints can be ordering constraints, resource bounds, or state limitations.

For the CX1 project demo, the model includes activity descriptions, uplinks, downlinks and engineering activities. The model also describes resources such as battery power, solar array power, and on-board memory. Periods of ground station visibility are modeled as states.

In Figure 5.10 we can see the ASPEN GUI in two phases of the CX1 demo. On the left side the initial state is loaded, and the figure on the right side shows the results after plan generation.

ASPEN's information visualisation approach is based on GUIs, where good use is made of graphical resources to represent activities against timelines with colour differentiation (for example, red is used to shown conflict in activities). However it is

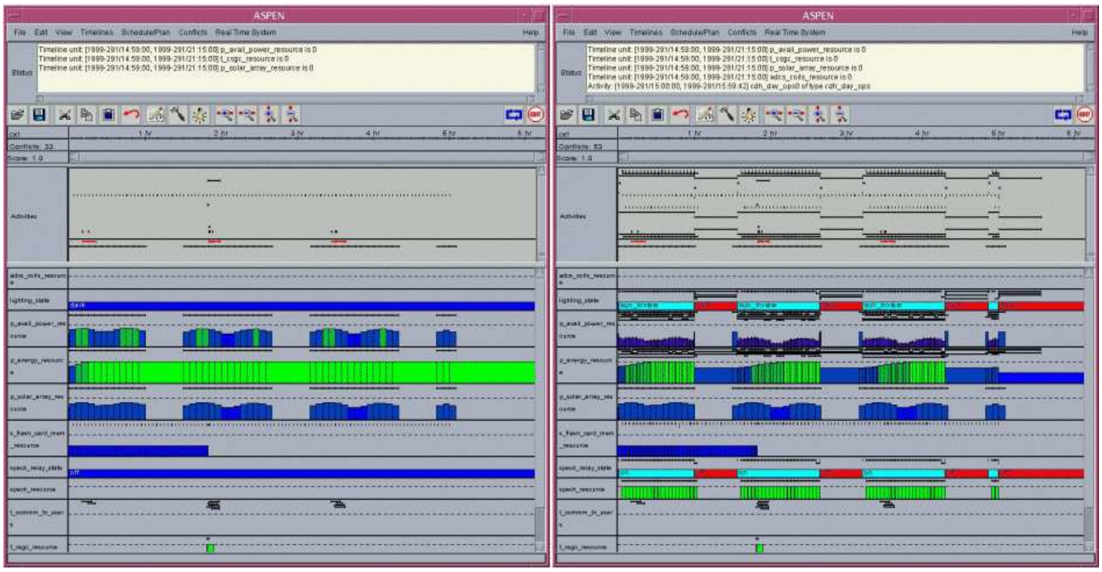


Figure 5.10: ASPEN GUI [Chien et al., 2000].

attached and dependent on the system approach: since it is also a scheduling system, the information visualisation method is strongly connected to the manipulation of time. In addition, despite the system having support for plan execution, and plan repair (by iterative repair), its user interface doesn't explore these aspects in depth. Plan repair is only supported when conflicts occur after plan generation, where conflicts can be repaired by making modifications manually using the GUI or by running the iterative repair algorithm. The plan repair algorithm can be invoked using the GUI or it may run automatically as conflicts arise.

5.3.8 MAPGEN

MAPGEN [Ai-Chan et al., 2004] is a mixed-initiative planning and scheduling project for the NASA Mars Exploration Rover Mission, launched in the summer of 2003. The objective of this mission is to elucidate the planet's past climate, water activity and habitability, using two NASA rovers - Spirit and Opportunity. MAPGEN is used as an activity-planning tool.

MAPGEN combines two existing systems: the APGEN [Maldague et al., 1997] activity planning tool and the Europa [Jonsson, 1999] planning and scheduling system. For each Martian day (sol) users on earth receive data from the rovers. Based on this data, they have to construct, verify, and uplink to the rovers a detailed sequence of commands to be used in the next sol that satisfies the mission goals. To help in this task MAPGEN can automatically generate plans and schedules, assist on hypothesis testing (what-if analysis on various scenarios), support plan editing, analyze resource usage and perform constraint enforcement and maintenance.

The planner's domain model specifies constraints (such as forbidden activity overlaps or resource violation). The model derives from an activity dictionary that describes abstract activities that the science user would need and flight and mission rules based on the project's flight rules dictionary. The planner's constraint engine enforces the domain model rules.

MAPGEN functionality is defined as follow: (1) during the activity plan-generation phase for uplink, science users construct a list of observation for each sol; (2) each observation consists of a collection of coordinated high-level activities; (3) APGEN expands these into lower-level activities based on the definitions in the activity dictionary. These activities together with the supplied engineering activities and initial conditions define the basis of the start of the planning phase.

The planner uses the domain model and generates a possible plan. The APGEN GUI then plays a role in assisting the mixed-initiative process of planning. The GUI displays this possible plan as a possible solution for the user to modify. MAPGEN also has another method that allows selective incremental planning of the high-level observation goals. In this method the user must determine the order in which observation goals are solved by selecting them in the GUI. In addition the user can experiment alternative *what-if* scenarios. Intermediate results are feed into the next iteration cycle in this mixed-initiative style until a final plan that the user finds appropriate is reached. When this process is completed the output is saved in a file for use in the next uplink process phase.

For the user input, there is also a separate tool, the *constraints editor* to enter the sol-based or daily constraints. This tool facilitates entering, visualisation and consistency checking of temporal constraints. After constraints are input via the GUI, the planner enforces these constraints to provide a more desirable solution according to the scientist user intent.

MAPGEN uses a concept of *flexible-time* to handle temporal constraints. It means that instead of finding a single solution, the planner preserves maximum temporal flexibility by maintaining a set of solutions that satisfy the constraints, represented internally as a *simple temporal network* (STN). However, representing such flexibility in the GUI creates problems. For any plan GUI, providing a visual representation of flexible windows, as well as binary temporal relations (such as before and after) it is difficult to find a suitable visual representation. APGEN as used with MAPGEN has certain tools, such as one for calculating resource usage that require a fixed schedule of activities.

MAPGEN solves this problem by presenting a single solution to the user in the APGEN GUI, while the planner maintains the flexible set of solutions as a backup. So, while the user sees a traditional fixed timepoint plan in the APGEN GUI, the underlying representation of the plan in the planner is a richer set of adjacent plans that all satisfy the constraint. Nevertheless, the user can also access the full set of solutions through *constrained move*. Constrained move consists of drag and drop activities moves by the user in the timeline. After this action of drag and drop, the planner adds a position constraint to fix it there. Such constraint is propagated to the other activities, which change their locations accordingly, by the minimum amount necessary to satisfy all the constraints. In this way the planner performs active constraint maintenance with minimum perturbation of the previous state.

In summary, MAPGEN enables visualisation and manipulation of plans for mixed-initiative interaction with the user, so that the MAPGEN GUI plays an important role in supporting mixed-initiative planning. To conclude this analysis of visualisation in planning systems, we can enumerate some well know planners that do not have user interfaces implemented. They are: Graphplan [Blum and Furst, 1997], TALplanner [Doherty and Kvarnstrm, 2001], MIPS [Edelkamp and Helmert, 2001], Blackbox [Kautz and Selman, 1998b] and FF [Hoffmann, 2001].

5.3.9 Sapa

Sapa [Do and Kambhampati, 2003] is a domain-independent heuristic forward chaining and temporal planner that deals with durative actions, metric resource constraints, and deadline goals. It uses a set of distance-based heuristics to control the search and can solve planning problems with complex temporal and resource constraints efficiently.

The Sapa action representation is mainly based on PDDL+ [Fox and Long, 2001] language, an extension of PDDL [McDermott et al., 1998] for expressing temporal domains. This representation permits not only the expression of instantaneous actions, but also actions with durations as used in temporal planners. It permits representation of: actions that have non-uniform durations, preconditions that are true at the start point or used to be maintained true for the duration of the action, and effects that are true at start or finish points of an action.

Sapa's search algorithm is a domain independent forward chaining heuristics and temporal planner. It does forward search in the space of time-stamped states. Sapa adapts the search algorithm proposed by [Bacchus and Ady, 2001]: a forward chaining algorithm that is able to use the type of action representation used in Sapa, and permits concurrent execution of actions in the plan.

To guide the search process, and cut out the bad branches early, Sapa uses heuristics. The heuristics used in Sapa are based on: a relaxed temporal planning graph, action durations and deadlines, efficient satisfying search, and metric resource constraints to adjust heuristic values.

The Sapa user interface is based on GUIs, and as it is a temporal planner, it gives emphasis to temporal information. The GUI is based on graphical charts. It permits visualisation of the plans generated by Sapa and relations between actions in the plan, for example, casual links, mutual exclusions, and resource relations. The charts used in the Sapa GUI have options to show: a time line of final plans (each action shown with its actual duration and starting time in the final plan), causal relations between actions, use of resources between actions, and also to illustrate specific times at which individual goals are achieved. Sapa GUI is shown in Figure 5.11.

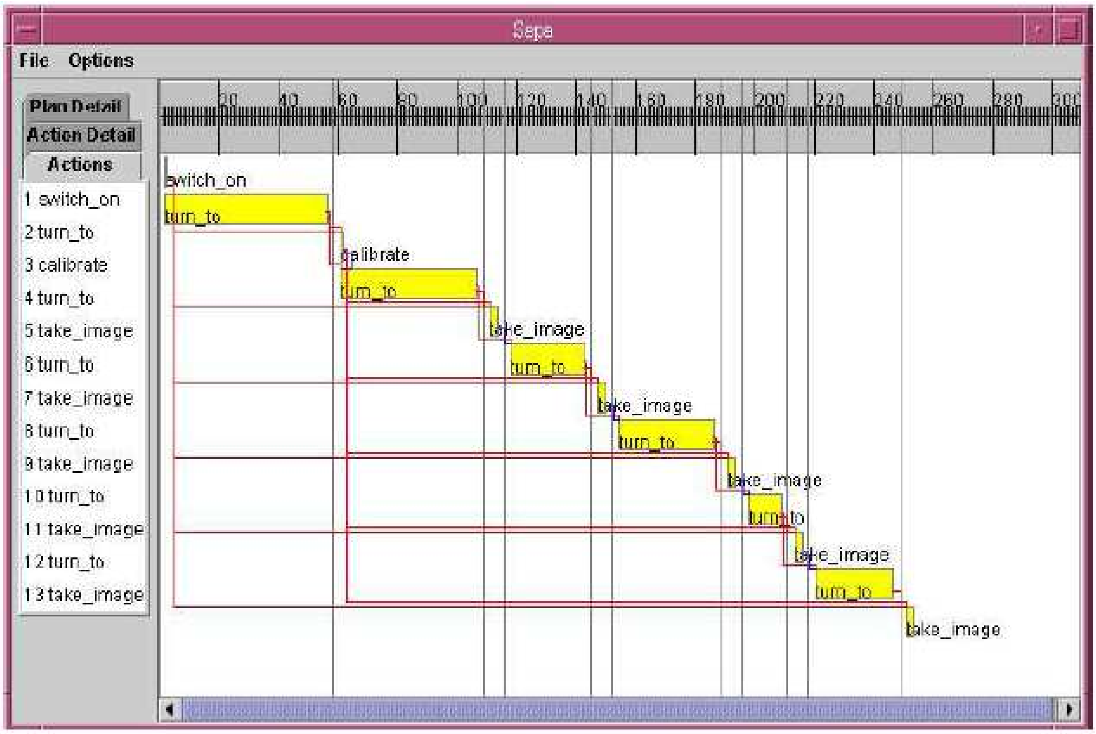


Figure 5.11: Sapa user interface - [Kambhampati, 2002].

Both planner and GUI are developed in Java. This aspect facilitates the development of other interface versions, such as a web one. So, Sapa also has a web-based interactive interface for the planner which has been developed as a Java Applet.

The Sapa GUI provides many technical details about the planning process that are useful for domains requiring a more complex notion of time. For instance, information is given for plan details (problem, functions, predicates, grounded actions, planning time, stages generated, stages explored), action details (index, start time, end time, duration, objects, pre action relations, post action relations) and plan domain.

However, for users without a background in planning technology the analysis of these temporal charts and information will not be an easy task. Another restriction is that the Sapa charts displays only the time line for the final plan. Intermediary or interactive results cannot be shown.

5.4 Categorisations and Comparison

Categorisations and comparisons of visualisation methods in intelligent planning systems can be made analysing them from different aspects and perspectives, such as:

- Which aspects of the planning process it supports:
  - Domain Modelling;
  - Planning Generation;
  - Planning Execution; and
  - Planning Simulation.
- Visualisation versus the planning approaches of search algorithm, plan representation and plan product.
- Related to the visualisation approach supported:
  - GUI: GUI approaches can be based on graphical resources of legacy systems or on new conceptualised and implemented graphical resources;
  - Natural Language;
  - Not Existent: some planners do not have a visualisation approach.

The investigation presented in this section tries to give a general notion about visualisation in AI planning.

5.4.1 By Visualisation Approaches

A general visualisation categorisation can be proposed based on the visualisation approaches used in intelligent planning systems. Table 5.1 presents the categorisation.

Category of Visualisation	**Planners**	**Advantages**	**Disadvantages**
External GUI	MPA, O-Plan	Lower implementation time cost Visually expressive	Lower customisation flexibility
Native GUI	ASPEN, PRODIGY/ANALOGY, Sapa, SIPE-2	Higher implementation time cost Visually expressive	Higher customisation flexibility
Natural Language Interface	TRAINS, TRIPS	Suitable for situations where visual interaction is not possible	Not where graphical representation has more expressive power
Without Implementation of Visualisation	Blackbox, FF/Metric-FF, Graphplan, MIPS, TALplanner	-	-

Table 5.1: Visualisation categorisation in AI planning systems.

The category *External GUI* includes the systems whose visualisation approaches are based on GUIs developed with legacy systems and/or off-the-shelf software. Examples of planning systems in this category are the MPA and O-Plan systems. In MPA the visualisation module is developed with legacy systems, and in O-Plan the GUI is implemented with off-the-shelf software, such as CAD systems.

The category *Native GUI* expresses the cases where a custom GUI is developed using the implementation platform. Planning systems in this category are: ASPEN, PRODIGY/ANALOGY, Sapa, and SIPE-2.

The category *Natural Language Interface* includes systems that communicate planning related information mainly via natural language processing technologies, instead of using graphical resources. The system TRAINS and its successor TRIPS are included in this category.

Others efficient planning systems do not have any implementation of information visualisation yet. The following planners are in the category *Without Implementation of Visualisation*: Blackbox, FF/Metric-FF, Graphplan, MIPS, TALplanner, etc. The number of planners that have not explored the visualisation aspect yet indicates the lack of research in this area, and the need of further investigation and improvements. The next section will discuss this need.

5.4.2 By Planning Aspect Supported

Different intelligent planning systems give support to different aspects of the planning process. Mainly, the user interfaces give more attention to plan generation. However, some planners also make efforts in domain modelling, plan execution and simulation user interfaces.

According to the planning aspect supported in the visualisation approaches, we can classify the support given in user interfaces approaches as:

- Domain Modelling User Interface Support;
- Plan Generation User Interface Support;
- Plan Execution User Interface Support;
- Plan Simulation User Interface Support; and
- Plan Repair User Interface Support.

Table 5.2 summarises the planning systems discussed in the previous section, according to the planning aspect that they support.

From this investigation we can note that the main focus of the user interfaces is actually in planning generation support. The approach most used for this is graphical, where GUI's are used to visualise planning information according to the planner paradigm and aims. So, for example, while the ASPEN plan generation GUI is very attached to the problem it solves (generate low level spacecraft commands from a set of high level science and engineering goals), the Sapa user interface puts emphasis

Planner	Domain Modelling	Plan Generation	Plan Execution	Plan Simulation	Plan Repair
ASPEN	-	X	-	-	-
MPA	X	X	-	X	-
O-Plan	X	X	X	X	X
PRODIGY	X	X	-	-	-
Sapa	-	X	-	-	-
SIPE-2	X	X	-	-	-
TRAINS/TRIPS	-	X	-	-	-

Table 5.2: Visualisation support in different planning aspects.

on temporal planning information. On the other hand, O-Plan adds the feature in its GUI interface of considering roles played by users in the planning process, and according to that, provides different views of plans. The PRODIGY planner adds elements of interaction in its user interface, where the user can play with the planning algorithm, following an animation, interrupting the processes for analysis, and changing search strategies. The SIPE-2 system also provides elaborated interaction permitting user input during the process, however some of the resources available require a strong planning technology background by users. Finally, among our analysed examples, the most distinctive approach is the one of the TRAINS/TRIPS systems which support plan generation through natural language processing techniques, providing a good level of interaction.

Regarding domain modelling support, these systems have different facilities for their user interfaces. MPA and SIPE-2 are both based on the Act Formalism for plan representation, and make use of a custom tool for domain modelling. There is an editing system for the Act formalism, called Act-Editor. The Act-Editor permits graphical browsing and editing for knowledge expressed in the Act language, and also of other resources, such as procedures. In addition, SIPE-2 also makes use of a generic knowledge base editor, the GKB-Editor, where a sort hierarchy can be created, viewed and edited.

O-Plan also provides an environment for domain modelling, using the Task Formalism (TF), a domain description language. TF is a framework for modelling and analysing planning domains. <I-N-OVA> and its successor <I-N-C-A> are used as the internal plan representation within O-Plan, where plans are represented as a set of constraints. Similarly, the PRODIGY interface also permits modelling plan domains, where it can be done in three different ways, from direct Lisp structure editing to graphical and form-based specification. Each way is more suitable or not depending on the user background on planning technology. On the other hand, ASPEN, Sapa, and TRAINS/TRIPS do not have support for knowledge input and editing.

Only some systems from our analysis give support to planning simulation and execution in their user interface. It is important to distinguish between having facilities for plan execution and giving support to that via a visualisation approach. On this basis, despite the ASPEN system allowing integration of planning and execution, its GUI does not give support for that. On the other hand, the O-Plan system provides an environment for execution of activity plans and it also has facilities that permit interaction with the system in a mixed-initiative style for simulation of plan execution and plan repair. In addition, its limited media interface approach includes a plan execution facility, where in order to execute a plan, the user is presented with a depth-first ordered list of nodes that have the status of being executable, given what has been completed so far in the execution process.

Similarly, there is little support to simulation in the user interfaces of the systems analysed. MPA provides some resources for plan evaluation and simulation output visualisation, while O-Plan World Views permits visualisation and creation of simulations and animations. Also, O-Plan permits interaction with the system in a mixed-initiative style for simulation of plan execution and plan repair. ASPEN, PRODIGY, Sapa, SIPE-2, and TRAINS/TRIPS do not present visualisation support to simulation. With the exception of O-Plan, very little support to plan repair is presented in the systems analysed.

5.4.3 By Search Algorithm, Plan Representation Application Domain and Visualisation Approaches

In this section different intelligent planning systems are analysed regarding their search algorithm, plan representation, application domain and visualisation approaches. Correlations were investigated, for instance, between search algorithms and/or plan representations and the visualisation approaches adopted by the systems. The following systems were analysed, each with different approaches for planning: ASPEN, Blackbox, FF, Graphplan, Metric-FF, MIPS, MPA, O-Plan, PRODIGY/ANALOGY, Sapa, SIPE-2, TALplanner, TGP, and TRAINS/TRIPS.

Table 5.3 and its continuation in Table 5.4 summarises the characteristics of planning systems in terms of search algorithm, plan representation, applicable domain and visualisation approach.

Analysing the data in Table 5.3 and Table 5.4 we can note that the approaches used for visualising planning information do not have any relation with (or influence by) the planning search algorithm used by the planners. However, the way that the plan representation is made may have an effect on visualisation modalities, since particular representations of plans can be more appropriate for graphical presentation than others.

Planner	Approach/Search Algorithm	Plan Representation	Domain	Visualisation Approach
ASPEN	Iterative repair	Constraint modelling language, expresses also temporal constraints	Space mission operations	GUI
Blackbox	Graph Based and Planning as Satisfiability-SAT	PDDL STRIPS	STRIPS benchmark problems (logistics, highly parallel domain, blocks world)	Not available
FF, Metric-FF	Heuristic Search	PDDL and ADL	STRIPS and ADL problems	Not available
Graphplan	Graph based	Planning Graph structure, STRIPS style language	STRIPS problems	Not available
MIPS	Binary Decision Diagrams (BDD) based	Binary decision diagrams (BDD), however receives as input language PDDL and ADL	STRIPS problems	Not available
MPA	Multi-agent	Act Formalism	Domain-independent (Applied in large-scale Air Campaign Planning domains)	GUI (Developed with legacy systems)
O-Plan	Knowledge-based and HTN	<I-N-OVA>/<I-N-C-A>	Domain-independent	GUI (Developed with off-the-shelf software and web)

Table 5.3: Part I - Categorisation by search algorithm, plan representation, application domain and visualisation approaches.

Planner	**Approach/Search Algorithm**	**Plan Representation**	**Domain**	**Visualisation Approach**
PRODIGY/ ANALOGY	Case Based and Generative	STRIPS style language, but extended for more expressiveness	Domain-independent	GUI
SAPA	Temporal Planner and Heuristic Search (Forward-chaining)	PDDL	Domain-independent	GUI
SIPE-2	Interactive	Act Formalism	Domain-independent	GUI
TALplanner	Handling Uncertainty, Temporal and Knowledge-based and HTN (Forward-chaining)	TAL (use of formulas in a temporal logic)	Domain-dependent (Uses domain dependent knowledge to control search)	Not available
TGP	Temporal and Graph Based	STRIPS extension for expressiveness	STRIPS problems	Not available
TRAINS/ TRIPS	Interactive Planning	Shared representation of plans	Logistic domains. Initially, routing and scheduling of trains. Later, evacuation domains (like Pacifica domain)	Natural language

Table 5.4: Part II - Categorisation by search algorithm, plan representation, application domain and visualisation approaches.

Chapter 6

Overview Summary

6.1 Problems and Gaps

This section discusses the existing problems and gaps in the area of information visualisation in intelligent planning systems, based on the analysis of the planning systems in the previous chapter.

Many advances have been made in intelligent planning systems, mainly related to the core problems, such as the development of faster search algorithms, finding the shortest plans, etc. However, there is a lack in research to provide better support for the proper use and interaction with planners. Only a few works address the problem of visualisation in planning systems. This problem is even more enhanced in collaborative planning environments, where visualisation can play an important role. There is a need for better support in collaborative planning systems as compared to planners working in isolation.

The main problems identified in this study of information visualisation in planning systems are:

- **Absence of solutions**: many successful and awarded planners do not even have a solution for information visualisation. The Graphplan [Blum and Furst, 1997] system is an example of such a planner. Despite its advances with respect to planning algorithms, this system does not have a way to communicate planning resources and output information to its users. Note, however, that Graphplan is an ongoing project, which is still the subject of research and improvements [Long and Fox, 2003]. This problem is also true for many other planning systems, as discussed previously.

- **Lack of flexibility**: some planners only consider one unique approach, when such an approach is not always appropriate for every situation. For example, the PRODIGY system which is mostly based on a GUI approach, while the conversational systems TRAINS/TRIPS provides a natural language approach.

- **Attention to a single aspect in the planning process**: visualisation approaches are generally destined only to address one aspect of the planning process, such as plan generation, leaving other aspects such as plan execution without support for information visualisation.

- **Software conception, architecture and lack of modularisation**: in many planning systems, even the ones with a modular approach, in general the visualisation software is designed with its features and/or modules only satisfying current requirements. However, if the planning system is subject to changes and improvements, the implementation that deals with visualisation will also have to be re-defined and re-implemented. In addition, if a system is built with similar planning concepts to another planner, the visualisation module in general cannot be re-used even when dealing with similar concepts, representations and consequently information. Thus, the way that information visualisation in planning systems is currently approached does not permit re-use of software. Hence, a new system or upgrade of systems implies requirements for a new visualisation module development.

- **Information Visualisation approach attached to planning paradigms**: the information visualisation approach is closely attached to one or more aspect of the planning paradigms adopted. This can be the domain of application, the paradigm/search algorithm used, the plan representation used, or the planner. For instance, if the system is about planning and scheduling for special domains, such as the ASPEN system, its information visualisation methods will have elements inherent to this domain as part of the core visualisation approach. Instead, to solve this problem, the information visualisation approach could have basic elements related to the planning approach, and customised elements that would permit it to deal, for instance, with application domain elements in a tailored way.

- **Lack of generality in the solutions proposed**: general visualisation mechanisms have not been proposed to deal with planning information and their use in practical and broad applications. Generality will increase, among other things, reusability and permit a more lasting approach.

The conclusion is that there exist plenty of research opportunities in this area. These research opportunities are discussed in the next section.

6.2 Research Directions

The existing problems and gaps in the area give rise to many research opportunities. Following there are some aspects that are investigated in the thesis and were taken in consideration when developing our solution:

- **Development of more general frameworks**: general frameworks will give support to different planning paradigms regarding information visualisation. This would permit a broader flexibility and increase usability and portability.

- **Use and integration of different modalities for information visualisation (multi-modal approach)**: the integration and use of different modalities of information visualisation (such as textual, graphical, natural language, virtual reality, etc.) will permit an appropriate use of each modality in different situations. Issues such as adaptation can also be dealt with. For instance, in a situation where the user is executing some task that does not allow him/her to pay attention to the screen (visual based mechanisms for information visualisation), sound can be used as an alternative approach.

- **Address issues regarding collaboration and different type of users involved in the process**: some situations and scenarios require collaboration between users to solve problems in a mixed-initiative fashion of planning. This leads to the question of different types of users (or human agents) taking part in the process. Human agents may have different backgrounds, capabilities, authorities and preferences when working in a collaborative planning environment. Thus, one direction of our research is to consider these questions in the context of visualisation to planning information.

- **Mobile computing for realistic collaborative environments**: information visualisation aimed at mobile devices can play an important role. In realistic environments human agents may need mobility to perform their tasks in the process.

So, the idea of delivering information to mobile devices can support the planning process in many ways, from generation to execution of plans.

All these points discussed above were considered and addressed in our approach and will be detailed in the next chapters.

To conclude, considering the lack of works in this area of visualisation of planning information and the new requirements of planning environments, such as realism and collaboration, there is a need to re-think the problem and consider the investigation of new and vanguard approaches. For instance, semantic based visualisation approaches, as they have already been considered as trendy approaches by the Information Visualisation communities.

Instead of an immediate solution for specific cases, it is necessary to globally comprehend the problem and associated elements, in order to represent the knowledge about this problem domain and permit general solutions and support for advanced services, such as an intelligent reasoning.

Part III

A General Framework for Visualisation in Collaborative AI Planning

Chapter 7

Framework - Semantic Modelling

This chapter introduces the framework proposed for semantic support for information visualisation in collaborative AI planning. The framework is divided in two parts: *The Semantic Modelling* and *The Reasoning Mechanism*. This chapter covers the first part, presenting the semantic modelling approach that consists of an integrated ontology set for describing and reasoning, in the context of collaborative AI planning environments. The general purpose of the framework is to provide a multi-modal way to support information visualisation in the context described. Nevertheless, the models can be used individually for other purposes.

The second part of the framework, regarding the reasoning mechanism, is presented in detail in the next chapter.

The general approach of the framework is proposed as a solution for organising and modelling knowledge related to a collaborative environment of planning, from an information visualisation perspective. In addition, it also permits embedded reasoning about the visualisation problem.

The remainder of this chapter is organised as follows. Section 7.1 introduces the general framework being proposed and the main ideas which underlie it. Section 7.2 introduces the first part of the framework regarding the semantic modelling and the approach of the ontology set. In addition, the section also discusses how the semantic modelling particularly fits this framework and the general role that it plays regarding semantics, knowledge representation and the Semantic Web. Section 7.3 goes into details of each developed ontology that composes this framework. Section 7.4 discusses the knowledge representation approach. Finally, Section 7.5 presents a summary about the whole framework, before a detailed explanation of the reasoning mechanism in the next chapter.

7.1 Introduction to the Framework

The main goal of this thesis is to propose a general framework for supporting information visualisation of planning information in a context of collaborative environments of AI Planning. The framework consists of two parts:

- **Framework Part I - Knowledge Representation**: considers the aspect of organising and modelling complex domain problems from the contextual information visualisation perspective; and

- **Framework Part II - Reasoning Mechanism**: is based on the semantic modelling of Part I, and gives support to reasoning about the contextual information visualisation problem.

Figure 7.1 illustrates the framework architecture. Using semantic modelling techniques (ontologies), several knowledge models complement each other to structure a collaborative planning information visualisation knowledge model.

This knowledge model framework permits modelling and organising collaborative environments of planning from an information visualisation perspective.

Based on that, a reasoning mechanism outputs information visualisation methods, tailored for each situation.

The semantic model of the framework is composed by the following (sub) models: (1) Visualisation Multi-Modalities, (2) Planning Information, (3) Devices, (4) Agents, and (5) Environment.

Formulating and giving context to the problem, the framework proposed is designed to support: (1) a collaborative fashion of planning, where human and software agents collaborate to solve problems; (2) using mobile computing when appropriate to deliver information to the collaborative users in several forms; and (3) consider a multi-modal approach for information visualisation.

It is important to note that the focus of the framework is in a *multi-modal* approach for visualisation of planning information. The option for multi-modal information visualisation is due to several factors, but mainly because the conceptual design of the framework is to be developed as a general solution, rather than attached to a particular way of visualising information, device display, capacity, etc.

To satisfy this requirement, the framework was developed including different modalities of information visualisation, and also, being able to be extended to new modalities.

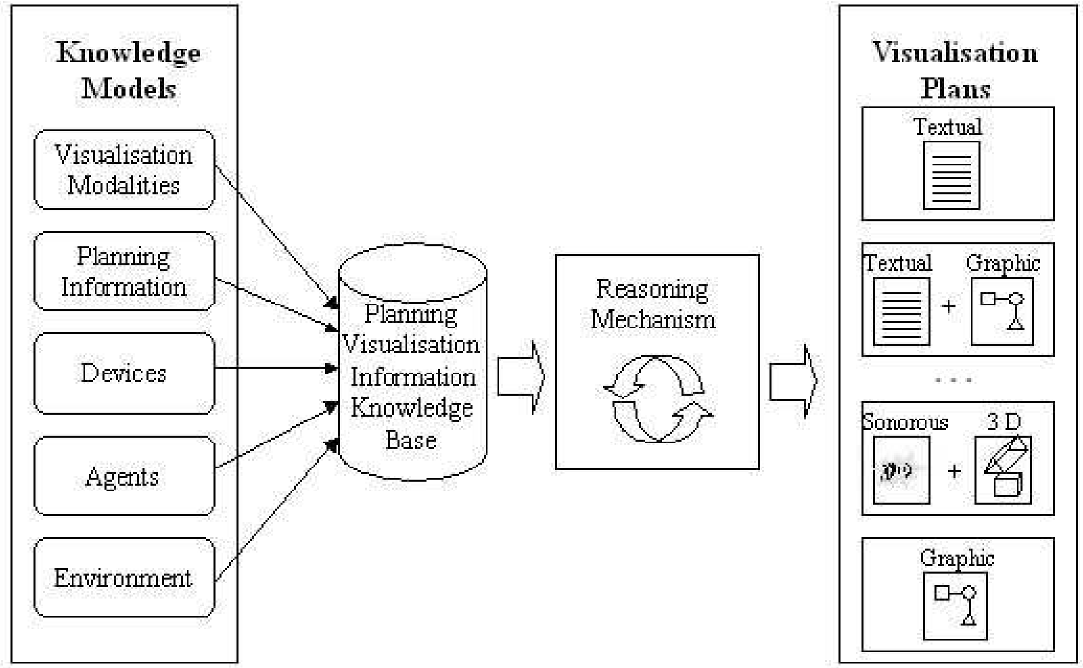


Figure 7.1: Framework architecture.

This will permit, for example, not only the integration of new technological innovations, but also the inclusion of new modalities suitable for new situations.

In the context of the thesis, the term multi-modality, or multi- modal is used to denominate conditions where two or more modes of operation exist. So the term refers to conditions where two or more forms of information delivery and visualisation are available and/or used. These forms are designed for both novice and expert users, and can be tailored to the needs of each user. Furthermore it can also means asking for some information in one form, and receiving in another.

In the last decade many works have been exploring the use of multi-modal information visualisation. A solution for multi-modality visualisation is proposed in [Moran et al., 1997]. In this approach, a multi-agent architecture, called Open Agent Architecture (OAA), is used to support multi-modal user interfaces. The Open Agent Architecture is a multi-agent system that supports the creation of agent applications, where part of the focus is on the user interface of such applications.

The supported interface modalities are: spoken language, handwriting, pen-based gestures, and Graphical User Interface (GUI). The user can interact using a mix of modalities. When a certain modality is detected by the system, the respective agent receives a message and processes the task.

The OAA has the following features:

- Open: It supports the integration of agents written in multiple languages (C, Prolog, Lisp, Java, etc.) and platforms (Windows, Solaris, etc.);
- Distributed: Agents in an application can run on multiple platforms;
- Extensible: Agents can be added at run time and their capabilities become available and also removed from the system;
- Mobile: Applications can be run from a mobile computer or PDA;
- Collaborative: There is no distinction in the interface between human and automated agents, which is claimed simplifies the creation of systems where multiple humans and automated agents cooperate. However there is no proper system giving support to collaboration processes;
- Multi Modalities: The user interface supports multiple modalities; and
- Multi Modal Interaction: User can enter commands with a mix of modalities.

The user interface is implemented with a set of agents controlled by an agent called *User Interface (UI) Agent*. This agent manages the various modalities. Thus, for example, it sends commands to agents to process, for instance, audio input; invoke agents that deal with the specific modality, when the UI agent detects a modality. The UI agent also produces a logical form of the user's request, that is passed to a *Facilitator Agent (FA)*.

The FA identifies the subtasks in the user request and delegates them to the appropriate application agents. In this way, the FA is the key for cooperation and communication between agents, since its job is to register capabilities of agents, receive requests and delegate agents to answer requests. The Facilitator Agent can be, however, a potential bottleneck, because it centralises the control of the application.

In addition, OAA gives support to collaboration, but it is not made in a very sophisticated way since there is no underlying system or mechanism giving specific support to collaboration. On the contrary, it is the done by the multi-modal user interfaces themselves.

In the recent past, sessions of the international conferences on Information Visualisation have been dedicated to the integration of visualisation and ontologies. Several works are proposing the use of ontologies in visualisation problems, and their application in the Semantic Web. References and discussions regarding these works can be found in Chapter 4, Section 4.5.

The approach proposed in this thesis has similarities and it is inspired by a mix of concepts from these works. It intends to be a multi-modality framework for information visualisation in a context of collaborative intelligent planning systems, based on industry standards (W3C [W3 Consortium, 2005a]) for semantic modelling and ontologies. This fact will permit the framework to work within the Semantic Web paradigm, allowing its easy use and application.

Regarding semantic modelling, ontologies have many advantages and disadvantages. Nevertheless they were chosen because they fit our requirements as stated bellow, and so far they have been the most used approach as a semantic modelling technique. The ontology community sees great potential in ontologies as an useful technology for building, manipulating, and reasoning on the Semantic Web and Semantic Grid. However, on the other hand, some in the planning community, are more sceptical and claim, for example, that ontologies are difficult to evaluate, mainly in a large context.

The question of why ontologies were chosen as a solution to be investigated is discussed bellow. We enumerate the following aspects about the advantages of using ontologies and why they fit our requirements:

- The integration of AI planning and ontologies (based on mark-up languages as knowledge representation tools) will permit the integration with the Semantic Web and Semantic Grid concepts. For instance, this integration will permit a broad application on the Semantic Web/Grid for visualisation aspects;

- Extensions of the framework will permit development of applications on the Semantic Web and Semantic Grid;

- AI planning technologies are already being used on the Semantic Web/Grid. For instance, in [Gil et al., 2004] a planning system is described to generate task workflows for the grid;

- The approach consists of a good opportunity for ontology based modelling from an information visualisation perspective in a collaborative planning environment;

- It is also a solution for the problem of ubiquitous and pervasive computing. It provides mobile devices with semantics and identity;

- Finally, ontology based semantic modelling allow us to add a level of reasoning about different aspects, such as, user's needs and preferences, adaptation related to services, user interface, etc.

The next section continues this discussion, giving more details about general aspects of semantics, knowledge representation Sematic Web and the relation of such concepts to our framework.

7.2 The Role of the Semantic Modelling Approach

The framework for information visualisation in collaborative planning systems is based on a semantic modelling approach. The investigation of this approach is based on some desired requirements for the final solution proposed, such as:

- The development of a general framework for supporting information visualisation in AI planning that would be independent of planners and their specific features. For instance, the internal representation used by a planner;

- Independence of existing and current technologies regarding visualisation devices. An approach based on semantic modelling and knowledge representation would allow attacking the problem from its conceptualisation, where not only can the modelling be done in a high level and abstract way, envisaging the advance in new technologies, but also, in knowledge base systems it is a relative easy task to extend the models to include new concepts and classes; and

- Considering current trends in Information Visualisation in using ontological based approaches. As discussed on Section 4.4.1, it is a current trend in Information Visualisation. In the most recent international conferences in the area, it has been witnessed that there is an increase in the communities interest in applying ontologies to information visualisation and their application on the Semantic Web. This aspect will give opportunities for a broader use of our framework.

In this context, the semantic modelling approach gives contributions at two levels:

- **Specific Level**: the contributions are regarding the modelling capabilities permitted by the models in the contextualised environment of collaborative AI Planning. The ontological representation/language allow us to describe the problem from the contextual information visualisation perspective. In addition it also permits the development of reasoning services based on contextual requirements;

- **General Concepts Level**: the contributions explore the potential use of the semantic modelling approach as related to knowledge representation, standardisation and Semantic Web/Grid concepts. Thus, at a broader level, the approach can fit into the Semantic Web concepts for the development of applications and standards. We argue that the models can be used individually, applied to other context problems, or grouped to problems related to AI planning on the Semantic Web/Grid, under the information visualisation perspective.

Knowledge representation and reasoning is the area of Artificial Intelligence (AI) concerned with how knowledge can be represented symbolically and manipulated in an automatic way by reasoning programs [Brachman and Levesque, 2004]. In order to contribute to intelligent behavior, knowledge representation focuses on knowledge. Humans act intelligently because they know many things and are able to apply this knowledge to adapt to their environments and achieve goals. Making an analogy, knowledge representation investigates what a computer agent needs to know to behave intelligently and what sort of computational mechanisms might allow its knowledge to be made available to the agent as required.

Knowledge representation and reasoning is the study of how knowledge can at the same time be represented as comprehensibly as possible and be reasoned with as effectively as possible [Brachman and Levesque, 2004]. Semantic modelling through knowledge representation languages and tools permits (via logic structures) to represent knowledge systems. Automated reasoning allows reasoning with these logical structures.

In recent years, semantic modelling languages and tools have been investigated for application in the Semantic Web. The Semantic Web is the extension of the current Web in which information is given a well-defined meaning, better enabling computers and people to work in cooperation [Berners-Lee and Miller, 2002].

The idea is to have an universally accessible platform that permits data integration, sharing and processing by automated tools as well by people. The Semantic Web infrastructure enables not only web pages, but also databases, services, programs, sensors, personal devices, and even household appliances to consume and produce data on the web. In the Semantic Web, a form of data integration is allowed by having data on the Web defined and linked. It permits effective automation, discovery, integration, and reuse across applications. This availability of semantic data on the web also gives a new dimension to software agents, permitting search, filter, transformation and use of information in new and existing ways.

The vision of the Semantic Web is that in the future it will provide interactivity in terms of collaborative tools and in real-time. Another interesting aspect of the Semantic Web that is relevant, in the context of the thesis, is the recent Mobile Web Initiative. The Mobile Web Initiative is about making it easy to make web sites which work on mobile devices, such as pocket PCs, PDAs, mobile phones, etc.

In parallel to that, the need for a broader use of knowledge-based planning has been discussed in recent years. In [Wilkins and desJardins, 2001] it is advocated that the use of knowledge-based planning will bring many advantages to the area, mainly when focusing on solving realistic planning problems. Complex domains can benefit from methods for using rich knowledge models. In this perspective, among the existing planning paradigms, hierarchical task network (HTN) is the most appropriate to this proposition. In contrast to methods that use a minimal knowledge approach, such as the ones that use the knowledge representation based on STRIPS. However, despite the advantages of the HTN paradigm, it also has limitations such as complete domain modelling, a very difficult task in real-world planning domains. Thus, there are many researches opportunities in order to improve and permit a broader use of knowledge models in real world planning problems.

According to [Wilkins and desJardins, 2001] and based on their experience in planning for military and oil spill domains, the following capabilities are needed to solve realistic problems: (1) numerical reasoning, (2) concurrent actions, (3) context-dependent effects, (4) interaction with users, (5) execution monitoring, (6) replanning and (7) scalability. However, the main challenges in real-world domains are that they cannot be completed modelled and consequently, they raise issues about planner validation and correctness. So, in order to make AI planning technology useful for realistic and complex problems, there is a need to improve the use of knowledge models in several aspects related to planning; and the development of methods and techniques able to process and understand these rich knowledge models.

Three types of planning knowledge are identified by [Kautz and Selman, 1998b]: (1) knowledge about the domain; (2) knowledge about good plans; and (3) explicit search-control knowledge. [Wilkins and desJardins, 2001] extended this list about planning knowledge mentioning that knowledge-based planners also deal with: (4) knowledge about interacting with the user; (5) knowledge about user's preferences; and (6) knowledge about plan repair during execution.

Recent research has been following these principles to develop more expressive knowledge models and techniques for planning. [McCluskey and Simpson, 2004], for instance, proposes a work from the perspective of knowledge formulation for AI planning, in a sense that it provides support for knowledge acquisition and domain modelling. GIPO (Graphical Interface for Planning with Objects) consists of a GUI and tools environment to support knowledge acquisition for planning. GIPO permits knowledge formulation of domains and description of planning problems within these domains. It can be used with a range of planning engines, since the planner can intake a domain model written in GIPO and translate it into the planner's input language. GIPO uses an internal representation that is a structured formal language to capture classical and hierarchical HTN-like domains. Consequently it is aimed at classical and hierarchical domain model types. The advantages of GIPO are that it permits opportunities to identify and remove inconsistencies and inaccuracies in the development of domain models and guarantees that the domains are syntactically correct. It also uses predefined "design patterns", that are called *Generic Types* and give a higher level of abstraction for domain modelling. GIPO has an operator induction process called *opmaker* which is aimed at a knowledge engineer who does not have a background in AI planning, to permit a successful use of AI planning paradigms. The GIPO plan visualiser tool allows engineers to graphically view the output of successful plans generated by integrated planners. However it assumes a domain knowledge.

Based on these discussions of knowledge enrichment in planning, this thesis argues that this vision should be even more augmented. Our claim is that knowledge enhancement can bring benefits to other areas related to planning, and we highlight the planning visualisation area. Knowledge models developed from the information visualisation perspective will permit modelling and reasoning about the problem.

Considering this wider background of knowledge, the semantic modelling approach gives contributions at a general level.

On the other hand, considering the contextual collaborative environment of AI planning under which the framework was developed and its objectives, this approach gives contributions at the specific level. The main contributions at the specific level are:

- The models/ontologies allow the description of the contextual environment of AI planning from an information visualisation perspective;
- The semantic-based framework gives support to a generic solution of information visualisation that, in particular, tries to be independent of specific technologies regarding planning systems (such as search approach, internal language/representation, output, etc.);
- The information visualisation categories taken into account in the approach try to be broad enough to fit different requirements and needs, and at the same time being independent of current technological limitations.

The main semantic modelling approach contributions at a general level are:

- Each model/ontology can be used individually for other purposes and needs. For example, the mobile devices model/ontology can provide description of limited resource devices for use in different applications, and with distinct alternative purposes;
- The models/ontologies individually and when combined have the potential for use and application in the Semantic Web and Semantic Grid.

In the next section, each of the models that compose the framework will be introduced and discussed in details.

7.3 The Semantic Modelling Approach and The Ontology Set Description

The semantic modelling concerns the following sub-ontologies:

- Multi-Modality Visualisation Ontology;
- Planning Information Ontology;
- Devices Ontology;
- Agents Ontology (Organisation and Mental States);
- Environment Ontology.

For the development of the ontologies, the concepts were based sometimes on existing models. In other cases the models were developed to attend the requirements of the problem that we are trying to solve. The next subsections describe the development, scope and main concepts of each ontology.

7.3.1 Multi-Modal Visualisation Ontology

The **Multi-Modal Visualisation Ontology** allows us to express the different modalities of visualisation considered in the approach. As the essence of the framework is to be generic, a broad range of modalities are considered.

The definition of this model is based on previous classifications of information visualisation categories existing in the literature [Card et al., 1999], while also trying to incorporate a diversity of modalities that will fulfill the framework's requirement of being general.

The model has three main concepts defined by the following classes (and their respective children in the class hierarchy): *Multi-Modality*, *Interface Component* and *Interface Operator*. The model explanation will be done by parts, according to these three main concepts.

Regarding the *Multi-Modality* conceptualisation, at the first level the information visualisation modalities are categorised into *simple structured* and *complex structured* classes. At the second level, however, the modalities are categorised according to their *dimensional representation*. At the final level, the modalities themselves are categorised. Figure 7.2 illustrates that (note that the other classes of the model were hidden here for legibility reasons). The complete model can be found in Appendix A.

In summary, the model contains the following modalities of information visualisation: *(1) Textual, (2)Sonore, (3) Tabular, (4) Graphical, (5) Map-Based, (6) Spatial Representation, (7) Virtual Reality, (8) Tree, (9) Network, (10) Temporal and (11) Natural Language*.

The definitions regarding the classes hierarchy shown in Figure 7.2 are described bellow:

- Multi-Modality: superclass of the model that involves all the possible ways of visualising/delivering information during the planning process;

- Simple Structure: the principal feature of this category is its easy way to be used. Generally it is based on a linear form of presenting information, so that

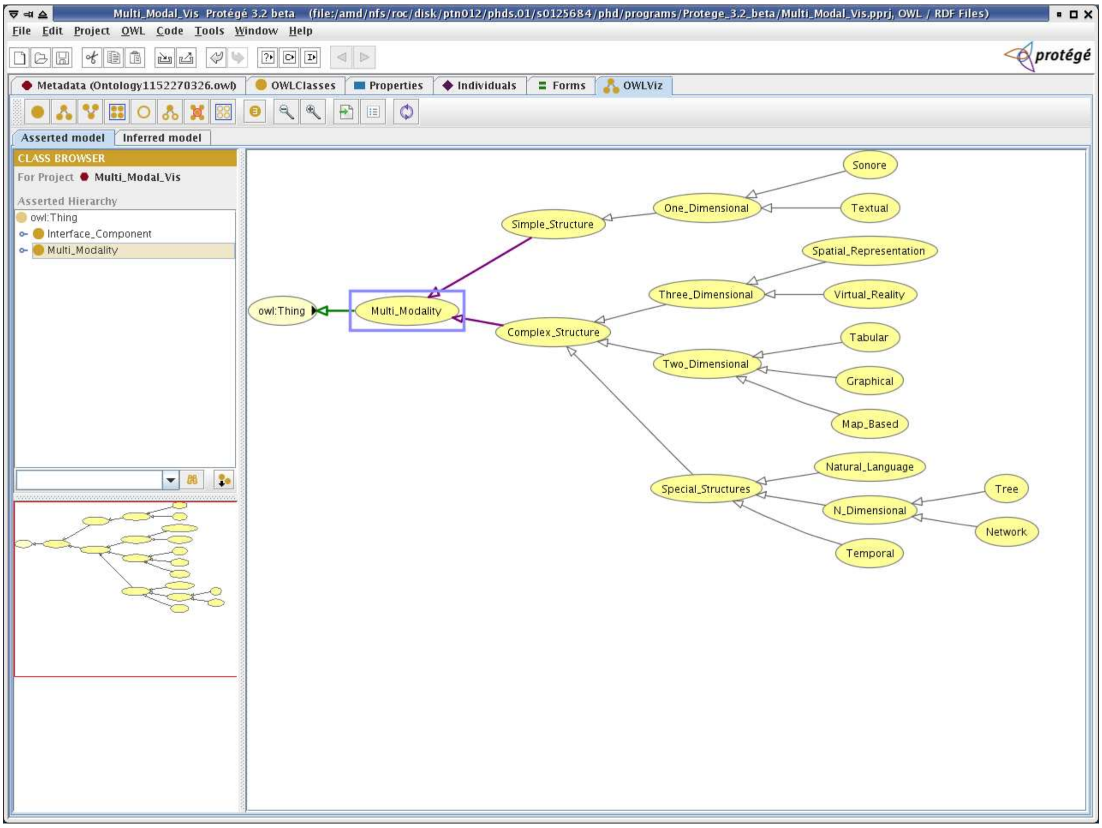


Figure 7.2: Multi-Modal visualisation ontology classes hierarchy - Multi-Modality conceptualisation.

their components do not support a direct way to present n-dimensional relations between two or more sets of information;

- One Dimensional: this category is related to forms of visualisation represented in one dimension;

- Text: category that represents textual information, which is typically composed from a sequence of symbols such as letters and/or numbers. Several aspects influence textual visibility, such as its length, colour, initial letter, spelling, etc.;

- Sonore: this group includes forms of visualisation/delivery based on audible information. The two subcategories here are: sound and voice based. Sound represents the category whose components are able to generate hearing information based on simple noises with some meaning. Note that here the information is not based on grammatical sentences, but on sounds like whistles. Voice is the category whose components provide a hearing way to deliver textual information. Note that we consider that such information is pre-recorded rather than generated at runtime. For example, messages used in lifts;

- Three Dimensional: represents categories that needs a 3D representation to better present information;

- Spatial Representation: category whose components are able to represent or compose information in three orthogonal axis. Examples are tri-dimensional plotters or representation of elements with volume;

- Virtual Reality: this category encloses the components that are synthetically able to create real environments, which represent a very complex way to relate information about objects and their behaviors;

- Complex Structure: this category involves more complex concepts that account for relating information in some kind of pre-defined structure. Such information comes from both the same or different knowledge groups;

- Two Dimensional: this category includes forms of visualisation represented in two dimensions. There are three subclasses of this category: tabular, map and GUI (Graphical User Interface) based;

- Tabular: category whose components display information as a crisscrossing grid of rows and columns. Each of the rectangles between grid lines, known as a cell, displays a value that in fact represents a relation between the row and column concepts;

- Map-based: category that defines components specialised in the representation of places and positions, as well as the relation between such concepts. Distance and scale are also important definitions of this class;

- The GUI-based contemplates more sophisticated graphical resources, such as icons and menus, instead of simple text for example;

- Special Structures: this category includes complex abstractions of data representation for information visualisation, such as natural language, multi-dimensional and temporal representations;

- Natural Language: natural language processing concepts are also considered in the semantic modelling. Natural Language technologies can allow, for example, the exchange of information between agents through conversational systems. Although it is claimed that natural language cannot completely substitute graphical interfaces [Shneiderman, 2000], it is suitable for many situations as it is going to be discussed later in the thesis. The components of this category are able to generate sentences of some language at runtime. As the information is generated during some process, it is more flexible and can relate data from different sources via structure of languages;

- N Dimensional: the multi-dimensional category concerns about representations considering more than three dimensions. One example of abstractions of this type is the use of parallel coordinates [Macrofocus, 2005] that represent several dimensions via a vertical bar for each dimension. Tree and Network visualisation are also included in this category;

- Network: category whose components are able to describe relations or connections among concepts or objects in a peer-to-peer way;

- Tree: category similar to networks, however that has as principal feature the representation of hierarchical information, where every concept or object has relations classified as superior (parent) or subordinate (child);

- Temporal: this category is concerned with conceptual notion of time information. Many solutions for temporal data visualisation is proposed on the literature. Temporal data needs a special treatment. For instance, works such as LifeLines [Alonso at al., 1998] addresses the problem. In the ontology, this modality abstracts the concepts involved in the presentation of temporal data.

Note that the way that the class hierarchy was organised for this ontology (Figure 7.2) means there is a unary branch: Simple-Structure $->$ One-Dimension. Despite the fact that unary branches do not really achieve much (being it plausible of elimination); its existence is justified by the reasoning mechanism. Some strategies of the reasoning mechanism consider the information regarding the complexity of the visualisation structures, i.e., if it is a simple structure or a complex structure. The reasoning mechanism is detailed in Chapter 8.

The second main concept to be visited in the semantic modelling is the *Interface Component*, whose class hierarchy is shown in follow (Figure 7.3).

This class (and its children) is related to the *Multi-Modality* class by the restriction *Multi-Modality hasComponent some Interface Component*. That means that an

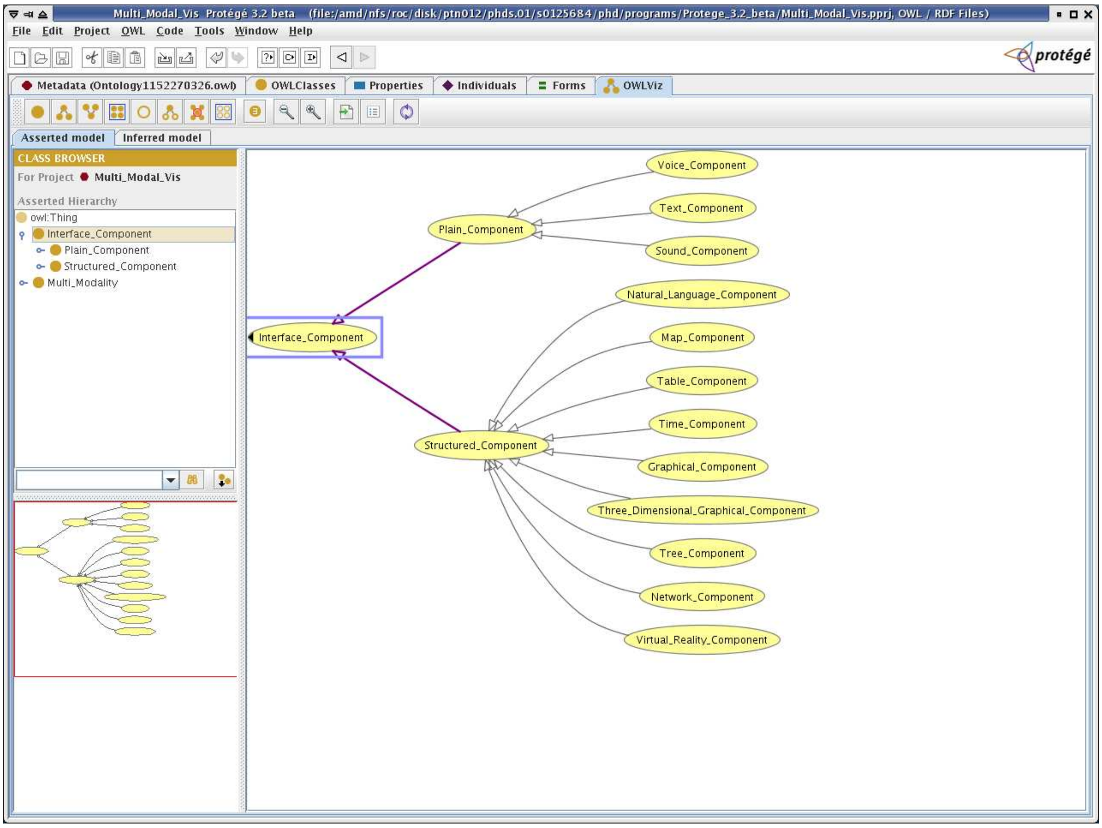


Figure 7.3: Multi-Modal visualisation ontology classes hierarchy - interface component conceptualisation.

instance of the Multi-Modality class has at least one (is related to) Interface Component.

For example, a textual modality of information visualisation would have text as interface component.

In other words, each of these components act as primitive elements during the creating of a specific interface.

The definitions regarding the classes hierarchy shown in Figure 7.3 are: Interface Component, Plain Component, Structured Component, Text Component, Sound Component, Voice Component, Table Component, Graphical Component, Map Component, Three Dimensional Graphical Component, Virtual Reality Component, Tree Component, Network Component, Time Component and Natural Language Component. The relation between these concepts and the visualisation modalities can be seen in Appendix A.

The last main concept to be discussed in the Multi-Modal Visualisation Ontology is *Interface Operator*.

This class (and its children) is related to the *Multi-Modality* class by the restriction *Multi-Modality hasOperator Interface Operator*.

That means that an instance of the Multi-Modality class has (is related to) Interface Operator.

For instance, a map modality of information visualisation may have zoom as interface operator, but not necessarily. This class hierarchy conceptualises the operations that can performed by the user in the information visualisation modalities.

The concepts definitions regarding the classes hierarchy of Interface Operator are described bellow:

- Obtain Details: select an item or group and get details when needed
- Extract: allow extraction of sub-collections and of the query parameters;
- Filter: filter out uninteresting items;
- Obtain History: keep a history of actions to support undo, replay, and progressive refinement;
- Overview: gain an overview of the entire collection;
- Relate: view relationships among items;
- Zoom: zoom in on items of interest.

These are the main concepts of the Multi-Modal Visualisation model/ontology. The entire structure of the model/ontology and the code specification can be found in Appendix A.

7.3.2 Planning Information Ontology

7.3.2.1 Ontology Specification

The **Planning Information Ontology** models information related to the planning process. It categorises, in a high level, planning information as of the following natures:

- Domain Modelling: this category includes concepts of planning information related to domain modelling, involving, for instance, description of goals, resources, etc;

- Plan Generation: here, the semantic modelling is concerned with plan generation information concepts and abstractions;

- Planning Execution: includes vocabulary regarding information on planning execution;

This ontology is based on <I-N-C-A> (Issues-Nodes-Constraints-Annotations) [Tate et al., 2003], a general-purpose ontology that can be used to represent synthesised artefacts, such as plans and designs, in the form of a set of constraints on the space of all possible artefacts in the application domain.

An illustration of <I-N-C-A> specification is shown in Figure 7.4. In this illustration we can see, among other elements, the four principal <I-N-C-A> components:

- Issues: state the outstanding questions to be handled and can represent unsatisfied objectives or questions raised as result of analysis or other deliberative processes;

- Nodes: describe components that are to be included in an artefact (in our case, in a plan). Nodes can themselves be artefacts that can have their own structure with sub-nodes and other <I-N-C-A> described refinements associated with them;

- Constraints: restrict the relationships between nodes to describe only those artefacts within the artefact space that meet the requirements;

- Annotations: account for adding complementary human-centric and rationale information to plans. In a general way, annotations can be seen as notes on plan components, such as nodes (activities) or issues, describing information that is not easily represented via the other <I-N-C-A> components.

Each plan represented via <I-N-C-A> is made up of a set of issues, a set of nodes and a set of constraints, which relate those nodes and objects in the application domain. Annotations can be added to the overall plan, as well as specifically to any of its components. Figure 7.5 shows the first level of the <I-N-C-A> specification for plans, where we can see the declaration for such elements.

The first part of the specification is dedicated to the declaration of variables. Variables are characterised by a unique identifier, a name and a scope (local or global). Local variables are only visible by the component that is using them. Thus, we can have, for example, several local variables with the same name in a plan. Differently,

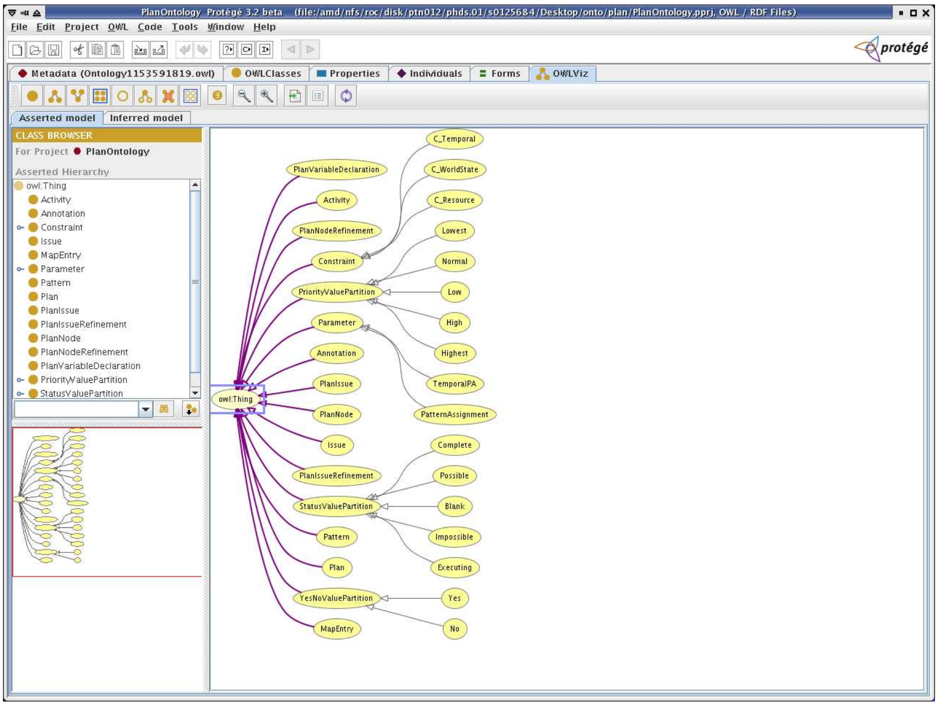


Figure 7.4: <I-N-C-A> specification.

global variables must have globally unique names. Names that represent variables begin with the symbol "?" and such names can be used by several other components of the model.

Issues, in this specification, are not directly included in a plan. Instead, each issue is wrapped in a PLAN-ISSUE element (Figure 7.6). A pair of the elements PLAN-ISSUE and PLAN-ISSUE-REFINEMENT is used to relate an issue to its sub-issues.

The ISSUE element (Figure 7.6) is characterised by a status (blank, complete, executing, possible, impossible, n/a), a qualitative priority (lowest, low, normal, high, highest), an attribute to indicate the source of the issue (sender-id), a reference name for internal use, and a flag to indicate if the issue sender requires report-back.

The declaration of nodes (activities) is similar to the issues, so that nodes are also not directly included in a plan. Using the same idea of issues, nodes are wrapped in a PLAN-NODE element and the pair of the elements PLAN-NODE and PLAN-NODE-REFINEMENT is used to relate an activity to its subactivities. Thus, the specification of the elements PLAN-NODE and ACTIVITY (Figure 7.7) are similar to the elements PLAN-ISSUE and ISSUE respectively. In fact, issues are likely to be transformed in activities during the planning process.

```
PLAN ::=
    <plan>
        <plan-variable-declarations>
            <list>PLAN-VARIABLE-DECLARATION</list>
        </plan-variable-declarations>
        <plan-issues> <list>PLAN-ISSUE</list> </plan-issues>
        <plan-issue-refinements>
            <list>PLAN-ISSUE-REFINEMENT</list>
        </plan-issue-refinements>
        <plan-nodes> <list>PLAN-NODE</list> </plan-nodes>
        <plan-node-refinements>
            <list>PLAN-NODE-REFINEMENT</list>
        </plan-node-refinements>
        <constraints> <list>CONSTRAINT</list> </constraints>
        <annotations> <map>MAP-ENTRY</map> </annotations>
    <plan>
```

Figure 7.5: First level of the <I-N-C-A> specification for plans.

A constraint (Figure 7.8) is characterised by a type (e.g., world-state), a relation (e.g., condition or effect) and a sender-id attribute to indicate its source. The constraint itself is described as a list of parameters, whose syntax depends on the type of the constraint. For example, a world-state constraint has as parameter a list of PATTERN-ASSIGNMENT, which is defined as a pair pattern-value such as ((speed wind),35km/h).

Finally we can see that annotations can be used in the high level plan definition, and also in each of its components. Annotations are represented by a set of key-value maps in which any object represented in the <I-N-C-A> specification may appear as a key or a value. The complete and current <I-N-C-A> specification for plans can be found in Appendix A.

7.3.2.2 Visualisation Process

The main focus of this ontology is to allow a generic conceptualisation of Planning information, so that the visualisation process can reason about the plan components

```
PLAN-ISSUE ::=
    <plan-issue id="NAME" expansion="NAME">
        <issue>ISSUE</issue>
        <annotations> <map>MAP-ENTRY</map> </annotations>
    </plan-issue>

ISSUE ::=
    <issue status="STATUS" priority="PRIORITY" sender-id="NAME"
            ref="NAME" report-back="YES-NO">
        <pattern> <list>PATTERN</list> </pattern>
        <annotations> <map>MAP-ENTRY</map> </annotations>
    </issue>
```

Figure 7.6: Specification of issues.

(activities, constraints, etc.) and decide on the best option to show this plan. The clear specification provided by <I-N-C-A> supports this process because the components are explicitly represented.

We consider that planning information can be used to meet different aims such as planning modelling, generation and execution. According to the literature and existing planning systems, planning information is approached in different ways, depending on the aim. So, delivering information for planning modelling is not the same as delivering for planning generation. Using <I-N-C-A> we can easily identify the plan components that are most related to the current aim. For example, if the system is in the execution stage, some important information to be displayed corresponds to the report-back of activities and their progress status.

Apart from the planning aim, it is possible to identify and classify planning information via the analysis of an instance of the model. For example, we can identify a group of temporal constraints, which have a different strategy of visualisation if we compare this with world-state constraints or a set of annotations.

All decisions based on a particular plan description will be performed by the reasoning mechanism (Chapter 8), which needs to present an understanding of planning information from a visualisation perspective. Note however, that such reasoning and decision making process is performed after considering all the context, which is modelled via the ontologies presented in this chapter.

PLAN-NODE ::=
<plan-node id="NAME" expansion="NAME">
<activity>ACTIVITY</activity>
<annotations> <map>MAP-ENTRY</map> </annotations>
</plan-issue>

ACTIVITY ::=
<activity status="STATUS" priority="PRIORITY" sender-id="NAME"
ref="NAME" report-back="YES-NO">
<pattern> <list>PATTERN</list> </pattern>
<annotations> <map>MAP-ENTRY</map> </annotations>
</activity>

Figure 7.7: Specification of nodes.

CONSTRAINT ::=
<constraint type="SYMBOL" relation="SYMBOL" sender-id="NAME">
<parameters> <list>PARAMETER<list> </parameters>
<annotations> <map>MAP-ENTRY</map> </annotations>
</constraint>

Figure 7.8: Specification of constraints.

7.3.3 Devices Ontology

In the 'Devices Ontology' [Lino et al., 2004] an approach for knowledge representation of devices capabilities and preferences concepts was investigated. We intend to integrate this into the framework proposed.

The CC/PP [W3 Consortium, 2004a] is an existing W3C standard for device profiling. The approach of CC/PP has many positive aspects. First, it can serve as a basis to guide adaptation and content presentation. Second, from the knowledge representation point of view, since it is based on RDF, it is a real standard and permits integration with the concepts of the Semantic Web construction. For our work, the Semantic Web concepts will also be considered. We envisage a Semantic Web extension and application of the framework that will be addressed in future publications. Third, another advantage of CC/PP is the resources for vocabulary extension, although extensibility is restricted.

On the other hand, CC/PP has some limitations when considering applying it to the realistic collaborative planning environment we are envisaging.

It has a limited expressive power, that does not permit a broader semantic expressiveness. Consequently it restricts reasoning possibilities.

For example, using CC/PP it is possible to express that a particular device is Java enabled. However this knowledge only means that it is possible to run Java 2 Micro Edition (J2ME) on that device. But, it can have a broader meaning, for example, when considering 'what really means to be Java enabled?' or 'what is J2ME supporting?'. Having the answers for questions like this will permit a more powerful reasoning mechanism based on the knowledge available for the domain. For instance, if a device is Java enable, and if J2ME is supporting an API (Application Program Interface) for Java 3D, it is possible consider delivering information in a 3D model.

For that there is a need to develop a more complex model for devices profiling that will be semantically more powerful. It is necessary to incorporate in the model other elements that will permit enhanced knowledge representation and semantics.

The 'Devices Ontology' proposes a new model approach that intends to enhance semantics and expressiveness of existing profiling methods for mobile and ubiquitous computing. Consequently, reasoning capabilities will also be enhanced. But, how the semantics will be improved? In many ways, as we will categorise and discuss bellow.

Semantic improvement can be categorised as follow in this new model being proposed:

- Java Technology Semantic Enhancement: In this category it is intended to enhance semantics related to the Java world. It is not sufficient to know that a mobile device is Java (J2ME) enabled. On the other hand, providing more and detailed information about it can improve device's usability when reasoning about information presentation and visualisation on devices. For that, this new model includes semantics for information about features supported by J2ME, such as support to 3D graphics; J2ME APIs (Application Program Interface), for instance, the Location API, that enables the development of location-based applications; and J2ME plug-ins, such as any Jabber [Muldowney and Landrum, 2000] plug-in available that will provide instant messaging, exchange of presence or any other structured information based on XML.

- Display x Sound x Navigation Semantic Enhancement: One of the most crucial things in the development of mobile device interfaces is the limited screen space to present information that makes it a difficult task. The two resources most used to bypass this problem are sound and navigation. Sound has been used instead of text or graphics to present information; for example, give sound alerts that indicate a specific message to the user. Indeed, it can be very useful in a situation where the user is on the move and not able to use hands and/or eyes depending on the task they are executing. Navigation can be used to improve user interface usability, if well designed. However, good navigation design has some complexity due to: devices diversity and because in some devices navigation is closely attached to the devices characteristics (special buttons, for example). So, this category intends to enhance semantics related to these aspects, that will permit good coordination and reasoning through these resources when presenting planning information to mobile device's users participating in a collaborative process.

- Open Future New Technologies Semantic Enhancement: This category of semantic enhancement is the most challenging one in the proposed new model. Mobile computing is an area that is developing very intensely. New devices and technologies are being created every day. In this way it's easy to create technologies that will be obsolete in few years time. Trying to overcome this problem, we envisage it will be possible to provide semantics to future new technologies in mobile computing via a general classes and vocabulary in the model and framework proposed.

In the next subsections an analysis of the CC/PP approach is made by a reverse engineering process, and consequently is discussed why CC/PP is not enough for what we envisage.

7.3.3.1 CC/PP Profiling: Reverse Engineering Analysis

Ubiquitous computing is an area that is growing very fast. Mobile devices are now everywhere and advances in wireless networking is making possible the development of more sophisticated applications. Nevertheless, the diversity of devices, technologies and applications available are making software development a difficult task, where applications have to be tailored for the different devices characteristics and capabilities.

In this scenario, devices profiling plays an important role. Profiling is one of the technologies emerging concerned with delivering content.

A device profile is a description of the device's characteristics in some way, which will guide content presentation. The World Wide Web Consortium (W3C) recommendation Composite Capability/Preference Profile (henceforth CC/PP) is one of these efforts developed to solve problems related to delivering content in devices.

A CC/PP profile is a description of device capabilities and user preferences. Resource Description Framework (RDF) [W3 Consortium, 2005b] is used as a knowledge representation tool to describe user agent capabilities and preferences, where RDF classes discriminate between different elements in a profile. CC/PP was chosen for grounding our investigation in device profiling for several reasons. First because it has an approach that best suits our concepts of knowledge representation. Second because it is based on W3C standards and concepts for the construction of the Semantic Web [W3 Consortium, 2005a], whose overall objective of enlarging the semantic web potential, reaching also mobile devices, is part of our global objective. At last, due to its popularity among mobile software developers, and use as a real standard. Hence, further investigation on CC/PP was carried out with the objective of identifying its expressive power as a knowledge representation tool. For that, based on the CC/PP RDF schema for classes and core properties, a reverse engineering process was applied. The main result of the process were a detailed UML class diagram. The class diagram helped identifying the CC/PP expressiveness: its scope, granularity of information, etc. The class diagram is illustrated on Figure 7.9.

A profile defines a document that describes the capabilities of a device to be exchanged between devices and guide content presentation. In particular, a CC/PP profile contains a number of CC/PP attribute names and associated values that are used by a server to determine the most appropriate form of a resource to be delivered to a client. Basically, the CC/PP vocabulary consists of a set of attribute names, permissible values and associated meanings. The CC/PP architecture is organized as follows: a profile is composed by one or more components, and each component has at least one or more properties. The classes that represent these main components in the class diagram are the classes CC/PP Profile, CC/PP Component and CC/PP Properties. The classes that compromise the CC/PP UML model and their description are listed bellow, however, for a better understanding the classes are grouped in accordance with their meaning and functionality.

Classes related to and which inherit from the RDF framework:

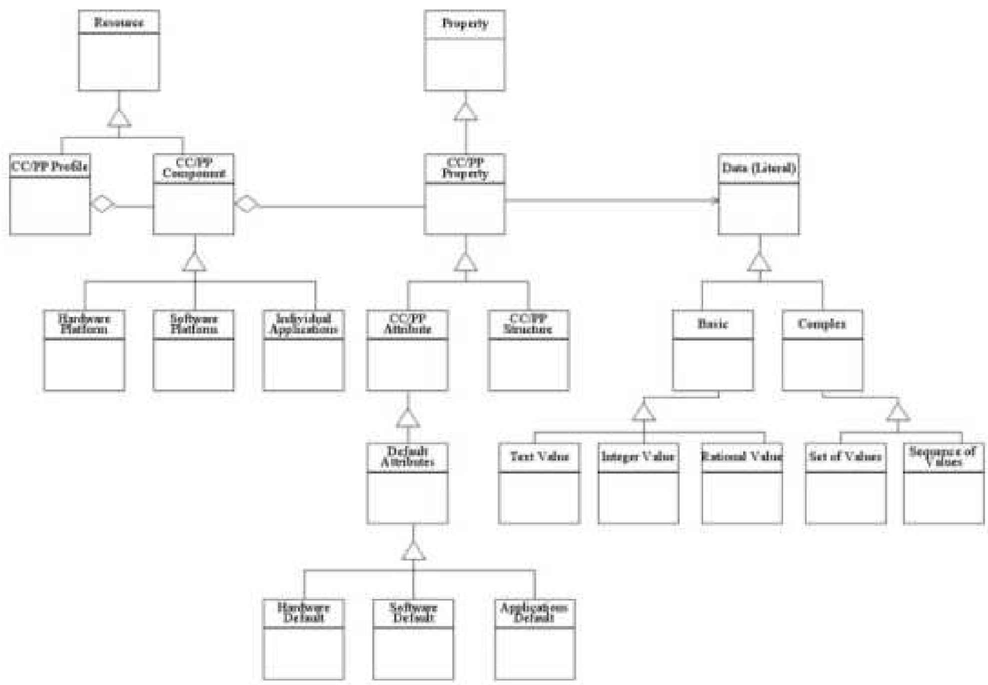


Figure 7.9: CC/PP UML class diagram created after reverse engineering.

- Resource;
- Property; and
- CC/PP Property.

These classes create the model due to the RDF philosophy for a general purpose metadata description language, which CC/PP is based on. However are not the core classes of the model.

The core classes of the CC/PP model are:

- CC/PP Profile;
- CC/PP Component;
- CC/PP Attribute; and
- CC/PP Structure.

These are the core classes of the model. A device profile is represented by the class CC/PP Profile, which is composed from one or more components. A component is an instance of the class CC/PP Component. Each component has one or more properties. All properties that are structural elements are defined as instances of CC/PP Structure, and all properties that describe a client device capability, characteristic or preferences should be defined as instances of CC/PP Attribute.

Component related classes:

- Hardware Platform;
- Software Platform; and
- Individual Applications.

In RDF notation, the definition of each component is a sub-tree, whose branches are the capabilities or preferences associated with that component. There are three groups of components: (1) hardware platform components, which contain for example display width and height properties; (2) software platforms components, where, for instance, operating system properties are specified; and finally (3) individual application components, containing properties related to user applications, such as the Mozilla browser.

Attribute related classes:

- Default Attributes;
- Hardware Default;
- Software Default; and
- Applications Default.

In order to minimize the use of the wireless network (and its limited bandwidth), the CC/PP profiling approach makes use of default attributes. Default Attributes are specified by reference to a default profile, which may be stored separately and accessed using its specific URL. It is a separate document that can reside at a separate location and can be separately cached. There are three classes of default attributes, all subclasses of Default Attributes: (1) Hardware Default; (2) Software Default; and (3) Applications Default, respectively representing default attributes related to hardware, software and user application properties.

Classes related to attribute values and data types:

- Data;

- Basic;
- Complex;
- Text Value;
- Integer Value;
- Relational Value;
- Set of Values; and
- Sequence of Values.

Finally, there are the classes abstracting the data manipulated by CC/PP and its data types. The class Data has two subclasses: Basic and Complex, representing respectively basic and complex data types. The basic data types include: text, integer and relational values, respectively instances of the classes Text Value, Integer Value, and Relational Value. In addition CC/PP also defines complex data types, for instance, set of values and sequence of values, represented by the classes Set of Values and Sequence of Values. A set consists of zero, one or more different values, where the order is not important. A sequence consists of zero, one or more values, where order is significant in some way.

Before discussing the positive and negative aspects of CC/PP, first it is necessary to explain why we are analysing it, and with which specific objectives. We are investigating an approach for knowledge representation of devices capabilities and preferences concepts that will integrate a reasoning mechanism of visualisation. That reasoning mechanism for visualisation is integrated from a collaborative intelligent planning environment perspective. It will deal with planning information and its tailored delivery and visualisation in different devices. Also, it has to consider collaborative users who are playing different roles when participating in a planning process. For that we need a powerful approach with great expressive power and flexibility. The approach of CC/PP has many positive aspects. First it can serve as a basis to guide adaptation and content presentation. Second, from the knowledge representation point of view, it is based in RDF, which is a good aspect because it is a real standard and also permits be integrated with the concepts of the Semantic Web construction. For our work, the Semantic Web concepts will also be considered. We envisage a Semantic Web extension that will not be treated in details here, put will appear in further publications. Third, another advantage of CC/PP is the resources for vocabulary extension, although extensibility is restricted. On the other hand, CC/PP has some limitations for what we need. It has a limited expressiveness power, that does not permit a broader semantic expressiveness. Consequently it restricts reasoning possibilities. For example, using CC/PP it is possible to express that a particular device is Java enabled. However this knowledge only means that it is possible to run Java 2 Micro Edition (J2ME) in that device. But it can have a broader meaning if we question, for example, 'What really means be Java enabled?' or 'What is J2ME supporting?'. Providing the answers to questions like that will permit a more powerful reasoning mechanism based on the knowledge available for the domain. For instance, if a device is Java enable, and if J2ME is supporting an API (Application Program Interface) for Java 3D, it is possible consider delivering information in a 3D model. For that is necessary to develop a more complex model for devices profiling that will be semantically more powerful. It is necessary to incorporate in the model other elements that will permit enhance knowledge representation and semantic.

Figure 7.10 illustrates our proposition for the devices ontology modelling. Like the other illustrations in this chapter, it is not the complete model. Some classes were hidden to prevent over cluttering of the information. The whole model is documented in Appendix A.

The new model approach for device profiling is motivated by the need for semantic enhancement to mobile device profiling. This work brings several contributions to the area. First it permits semantic improvement related to Java technology. This will allow reasoning considering Java aspects (resources, API's, plug ins, etc.) enabling the reasoning mechanism to propose tailored modalities of information visualisation. Second, it is also being provided semantic enhancements related to display, sound and navigation aspects, motivated by the fact that a wise use of these resources can improve mobile devices usability. Additionally, the most challenging contribution is that the approach does not intend to be limited to current technologies, but is open and extensible to new technologies semantic formatting.

7.3.4 Agents Organisation and Mental States Ontology

The main requirements of this model/ontology is to satisfy needs for reasoning about agents (software and human) roles in the organisation when participating in collaborative processes of planning, and all aspects related to it. In addition, also the agents men-

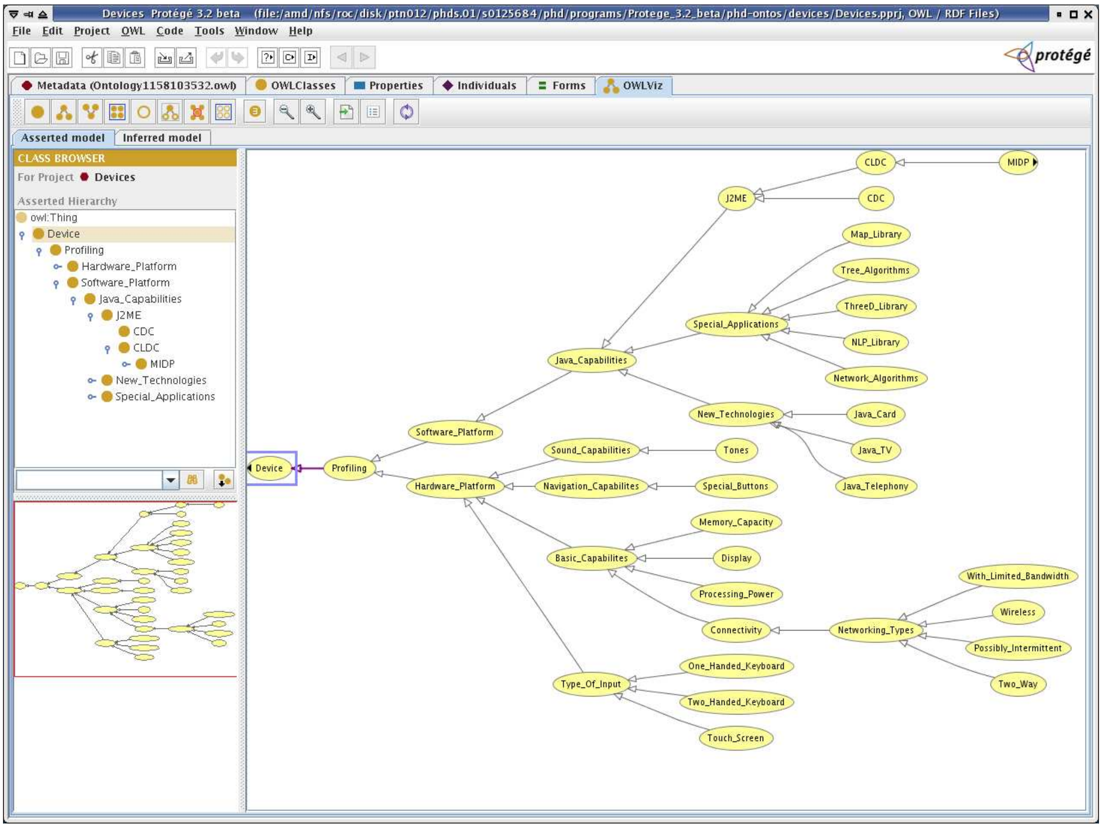


Figure 7.10: Class diagram of the devices ontology.

tal states regarding their goals, strategies and preferences in the process. The concepts modelled in this ontology and how they influence in the visualisation are discussed as follows:

- Mental States: describe the agents via concepts like goals or intentions, beliefs, commitments and desires Such concepts have a direct relation to the planning process and must be considered during the visualisation of plans. For example, intentions are similar to the idea of activities, already discussed in the planning ontology;

- Roles: this concept has to do with the role the agent plays in the planning process. Roles are also associated with responsibilities, capabilities and authorities. Depending on the role that agent is playing, there are more important or appropriate sets of information that this agent must focus on. Thus, roles can be understood as a filter of information and, consequently, this concept has influence on the visualisation;

- Relationships: agents are organised in virtual arranged in an organisation in some kind of structure, such as a hierarchy. In this way, agents are related to each other via some relation, as for example superior, subordinate peer or contact. Relationships define rules regarding the interaction of agents (e.g., delegation of tasks), which should be represented via some visualisation strategies;

- Preferences Profiling: through the concepts modelled here, agents can specify preferences regarding modality of visualisation, devices properties, etc. Based on these profiling, it is possible the adaptation of planning information presentation and delivery to the agent requirements.

The development of this ontology is based on two existing model concepts: BDI [Rao and Georgeff, 1995] and I-Space [Tate et al., 2003]. BDI (Belief-Desire-Intention) is the most popular concept used in the agent-based modelling and programming. In BDI, B stands for Believe (Data), D represents Desire (Goal) and I stands for Intention (Plan). Each agent has its own BDI model and, in order to achieve some goal (Desire), the agent can analyse its related data (Belief) and choose an appropriate plan (Intention).

The I-Space approach supports the arrangement of coalitions, allowing the management of organisational relationships such as superior-subordinate or peer-peer. Considering an agent *ag*, I-Space shows the kind of relationship that *ag* has with other agents of the coalition (superior, subordinate or peer). For each of these relationships we can associate specific forms of interaction, which characterise each relationship specifically. In addition, I-Space also shows the capabilities of each agent that composes the contact list of *ag*.

Based on these two concepts, BDI and I-Space, the agent model for planning visualisation is illustrated in the following (Figure 7.11):

The entire model with its classes and subclasses is presented in the Appendix A.

7.3.5 Environment Ontology

The environment ontology is responsible for permitting the expression of environment awareness. In particular, location based awareness is being considered, where this kind of information can be based on GPS (Global Positioning System) or any other location system. The idea considered in this section is that some features of the environment can have an influence on or guide the form of visualisation, so that such features also need to be semantically modelled in our representation.

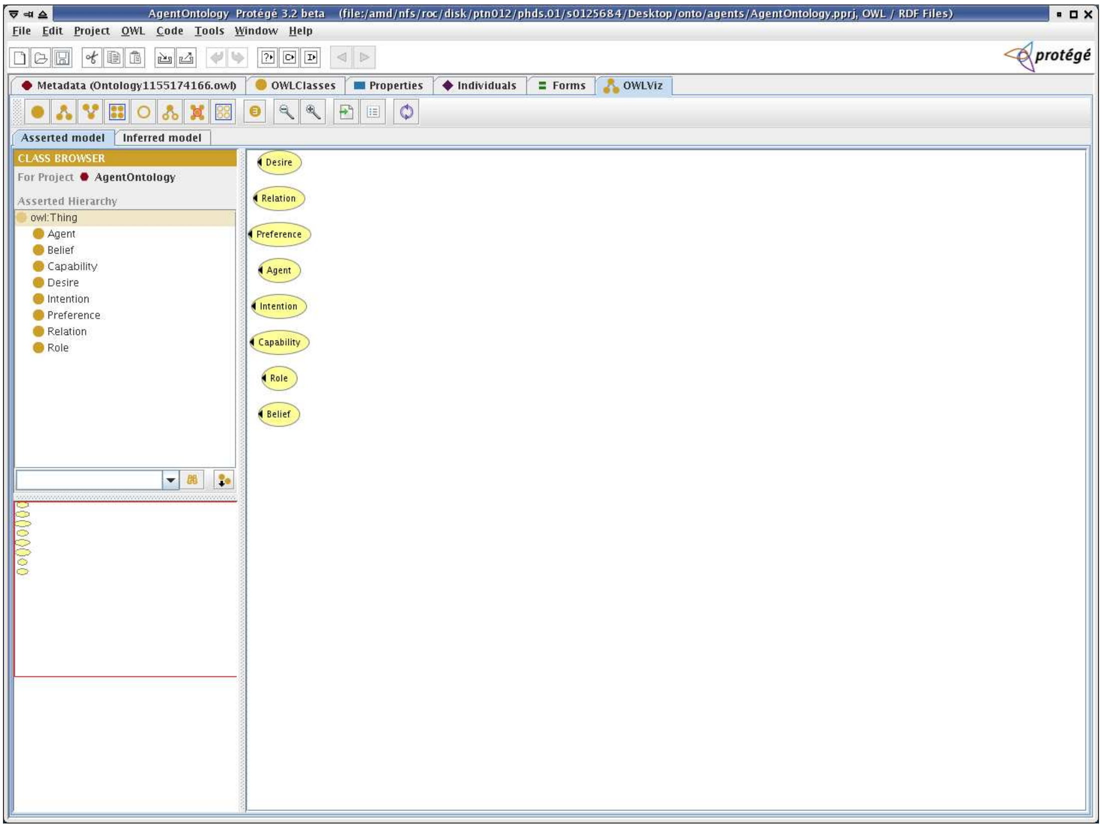


Figure 7.11: Class diagram to the agent ontology.

The main concept modelled in this ontology is Geographic Location. According to our model, every environment should have a location system that identifies the position of agents and objects in such an environment. Figure 7.12 presents this concept.

According to this model, every environment has a position system, which can be one of four subclasses: GPS, reference-based, descriptive or special. GPS gives the location of objects via the latitude and longitude attributes. In addition we can also consider altitude as a non-compulsory attribute to this system. Note that position systems based on latitude and longitude are not exclusive to Earth, so that it can be used on any planet. The difference will be the degree/distance relations which have a specific value depending on the circumference of each planet.

The reference-based system gives the position of every object in the environment as the orthogonal distance (axis x, y and z) between this object and a referential and generally fixed point. This system assume some metric unit, such as metre, associated with these distances.

The descriptive system is represented by a natural language description of a po-

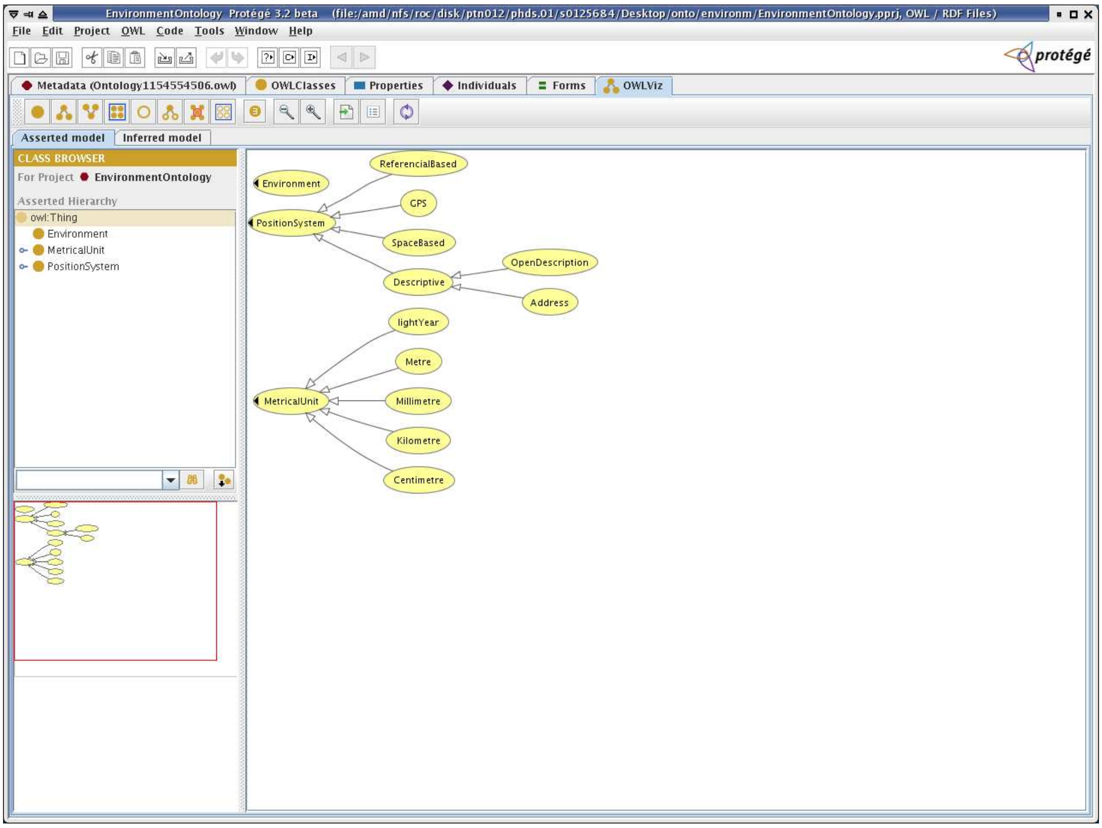


Figure 7.12: The position system model in the environment ontology.

sition or place. This category can be decomposed into two subclasses: formal and informal descriptions. The formal description is mainly represented by addresses. Addresses have attributes (e.g., road, number, postal code, etc.) that together indicate a specific position inside the environment. However, this representation is very limited because it does not cover all the positions as a latitude/longitude representation can do. The informal description does not have a pre-defined format and can look like: *I am in the Highlands on the West shore of Loch Ness, four kilometers South of the Urquhart Castle*.

Special location systems are associated with environments where the representation of objects are given in a more complex way. Deep-space exploration missions are examples of domains where the environment, in this case space, does not have a common way to represent positions of its objects. Thus, different approaches for each case must extend this class to define appropriate location systems.

It is important to note how different location systems can influence the visualisation decision process. Consider, for example, the use of a referential-based system during a rescue operation inside a big building such as a tower with several levels. For this scenario a 3D representation is the most appropriate due to the importance of the three dimensions during the navigation inside this building. In another example where we have an informal description of a location, a textual visualisation could be a simpler way to deliver this information. Due to the fact that the reasoning cannot place this position in a map or any other visualisation resource.

The representation of information associated with position systems is defined in our approach as instances of the <I-N-C-A> world-state constraint (Figura 7.13).

```
KNOWN-CONSTRAINT ::=
    <constraint type=world-state relation=effect>
        <parameters> <list>PATTERN-ASSIGNMENT<list> </parameters>
    </constraint>
```

Figure 7.13: Specification of world-state constraints.

Now we need to define the PATTERN-ASSIGNMENT (attribute object = value) to each position systems subclass (GPS, reference-based and descriptive). The table[1] in follow (Table 7.1) shows this definition.

	Attribute	Object	Value
GPS	latitude	?object	?lat
	longitude	?object	?lon
	altitude	?object	?alt
Reference-based	xReference	?environment	?xr
	yReference	?environment	?yr
	zReference	?environment	?zr
	x	?object	?x
	y	?object	?y
	z	?object	?z
Descriptive	address	?object	?addr
	positionDescription	?object	?posdescr

Table 7.1: Pattern-assignment to position systems.

[1]Question mark in front of any element means that such a element is a variable.

Note that all descriptions and specifications that we have done until now are associated with the position system. However several features of the environment can be described using this generical constraint template together with its pattern-assignment element. Furthermore, it is important to stress that we are not trying to model all the environment. This ontology should be a subset of a complete ontology for the environment, so that it only considers the features that have some kind of influence on the visualisation reasoning process.

7.4 Discussions Regarding Knowledge Representation Approach

The knowledge representation approach that we are investigating (OWL)is based on XML - Extensible Markup Language [W3 Consortium, 2005c] and related technologies, following the W3C [W3 Consortium, 2005d] standards.

Initially, XML related technologies are used as knowledge representation tools, however a Semantic Web [W3 Consortium, 2005a] application will not be aimed at first. These technologies filled a gap, providing first a syntax for structured documents (XML, XML Schema), and second a simple semantic for data models (RDF - Resource Description Framework), that evolved for more elaborated schemas (RDF Schema, OWL). RDF Schema permits semantics for generalization-hierarchies of properties and classes. OWL - Web Ontology Language [W3 Consortium, 2005e], adds more vocabulary with formal semantics, allowing more expressive power, permitting, for example, express relations between classes, cardinality, equality, and characteristics of properties.

OWL is an evolution of DAML+OIL [Horrocks 2002] and is intended for use when it is necessary to process information, not only present it, because it facilitates machine interpretability via its additional vocabulary and formal semantics. OWL is divided into three sub-languages, with increasing expressiveness: OWL Lite, which provides a classification hierarchy and simple constraints; OWL DL which has maximum expressiveness with computational completeness and decidability, founded by description logics; and OWL Full which allows maximum expressiveness and syntactic freedom of RFD, but without computational guarantees.

The OWL ability for processing semantic information seems to be an appropriate technology to be used in the general framework being developed, to build the integrated ontology set, and reasoning mechanism in the problem domain. The resulting framework will considers the semantic of the information available, and it will be capable of reasoning based on real standards.

An important aspect to consider, however, is that the use of W3C standards does not necessarily mean a Semantic Web application. Nevertheless, it is intended that a further investigation of the framework extension, will allow its application on the Semantic Web. One opportunity is to provide mechanisms for automatic semantic knowledge bases updates. For example, directions can be formulated to build agents for mobile devices update in a knowledge base.

7.5 The Big Picture - The Framework Summary

After that explanation about our representation approach, we give an overview about our framework here. To that end, this section presents the principal components and concepts of the framework, organising the discussion into the following topics: global architecture, types of users involved, different systems involved, types of knowledge/data representation used and mappings between them.

7.5.1 The Framework Overview

As introduced in the previous sections of this chapter, our framework is divided in two parts: the Semantic Modelling and the Reasoning Mechanism. Figures 7.14 illustrates, at a high level, the global architecture of the framework. Based on this figure, we present a summary of the framework, regarding both parts (Semantic Modelling and the Reasoning Mechanism).

Many elements are included in Figure 7.14, such as: types of users, methods of data and meta-data representation, processes involved and outcome. We give a glance at them here.

The framework architecture (Figure 7.14) shows the roles that users play when interacting within the framework, in both the Semantic Modelling and Reasoning Mechanism phases. The role of the Model Designer user is to develop the conceptual models. To that end we used Protege as a tool to create and edit OWL ontologies. For instance, the Model Designer is able to extend the models to express new concepts required for new situations. In addition, this user should also anticipate eventual need for new rules required by new conceptual models.

The responsibilities of a Domain Specific Designer are to create instances of the ontologies, according to the conceptual models developed by the Model Designer user. Therefore Domain Specific Designer must have domain expertise to carry out this role. Note that these two user types are involved in the Semantic Modelling phase of the framework.

Collaborative Agent users interact with the framework in the Reasoning Mechanism phase. In this way, they are the end users of the framework, so they do not worry about the engineering work behind it. At a high level, when Collaborative Agent users interact with the framework, all the reasoning process occurs (based on the semantic modelling), producing information visualisation solutions/recomendations as result, according to the scenario definition and the context where the agent is inserted into the collaborative process.

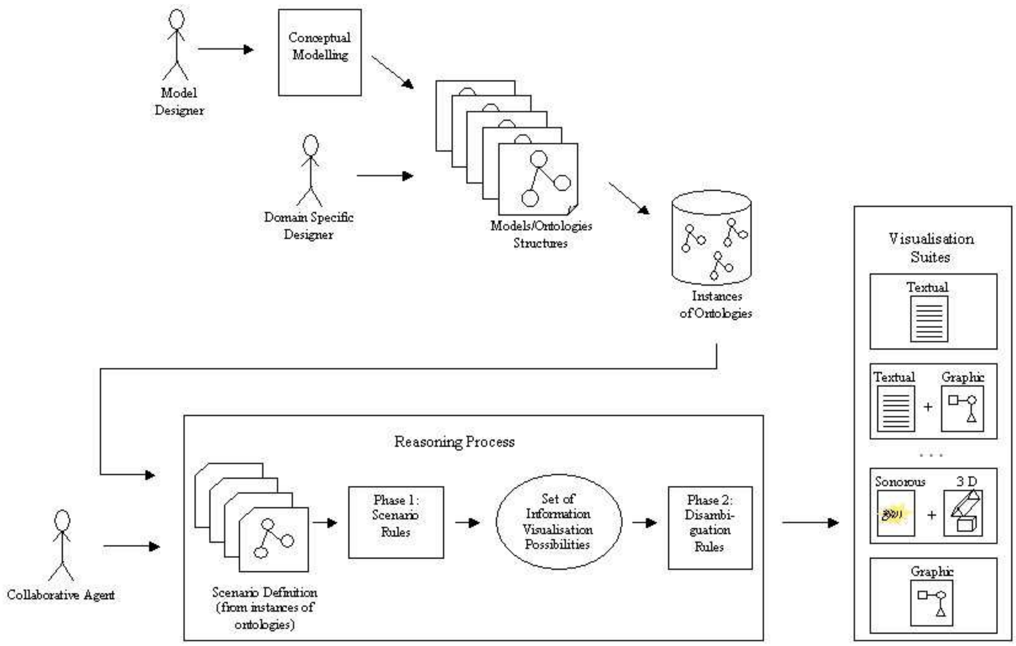


Figure 7.14: Framework architecture - general view.

Different processes that compose the framework are also illustrated in Figure 7.14. As our aim here is to give an overview of the framework, we simplify the discussion by illustrating only the principal processes. Such process are: the Conceptual Modelling, the Domain Designing, the Reasoning Mechanism and Generation of Multi-Modality Recommendation.

The Conceptual Modelling, or the Semantic Modelling process, was carried out in this thesis. It consists of the development of models according to the requirements and goals we intended to achieve. A set of five models were developed to support the information visualisation approach that we are proposing. Because this framework is extensible, additional conceptual modelling processes could be necessary to semantically express new extensions.

During the Domain Designing process are created instances of: users that are going to act on the environment, devices that are going to be employed by these users, the environment of performance and possible pre-defined plans called *Standard Operating Procedures* (SOPs). However, the majority of plans are created at runtime based on operators which must also be defined during the domain design process.

The reasoning mechanism process acts on instances of a specific scenario definition, returning all possible visualisation modalities for this scenario. In an optional second reasoning phase, disambiguation rules can be applied to the first phase outcome, filtering the results so that just one form of visualsation is chosen in the end.

The Generation of Multi-Modality Recommendation process is responsible for the ultimate (but not only) outcome of the framework, which is the recommendation of Multi-Modal information visualisation in contextual environments of collaborative planning. These recommendation are implemented in the form of visualisation suits that can be chosen by user agents to visualise information.

Chapter 8 brings details of these processes and others that are not cited here.

7.5.2 Users

In brief, users that interact with this framework can be classified according to their roles. Such main roles are:

- Model Designer: this user accounts for the conceptual design of the models. This means possible extensions to the semantic model itself. Furthermore, this designer must also keep the rule base updated in accordance with such extensions. Note that the current model offers a basis that can be augmented, according to new definitions and requirements;

- Domain Specific Designer: this user is a specialist in the concepts of the domain that is going to be modelled. This designer does not specify new model classes. Rather, he/she instantiates such classes to create the specific components required by a domain or application;

- Collaborative Agent: this is the final user of the system. Such users abstract all the technical details of the framework, so that they are using the facilities that such a framework brings to their collaborative planning tasks.

Note that we are supporting a wide range of users that can interact with the framework, carrying out tasks of extension, instantiation, or carrying out some collaborative planning task.

7.5.3 Systems

The framework uses some systems and technologies, which support the mapping from theoretical concepts to a real practical system. The use of such systems are detailed in Chapters 8 and 9, however we give a glance at them here:

- Protege [Knublauch et al., 2004]: is used as a specification environment to design ontologies in the OWL [W3 Consortium, 2005e] language. We used Protege due to the features and facilities it offers that satisfies our needs. For instance, Protege has a graphical and OWL environment for the creation of ontologies, and a large number of plug-ins. However other modelling tools could be used in the conceptual modelling, as long as it can produce a pure OWL file as output of the modelling;

- RACER [Haarslev and Moller, 2003]: is used as an inference engine for the ontologies developed in OWL. It is mainly used to check consistency and structure of such ontologies;

- Java [Sun Microsystems, 2006]: is used as the core language to integrate different components of the system, such as the I-X [Tate, 2001] system (also developed in Java) and the ontologies and their instances in JEOPS representation;

- JEOPS [Figueira and Ramalho, 2000]: the *Java Embedded Object Production System*, is used as inference engine, which uses rules in the JEOPS format to reason on facts specified according to the ontologies. Note that while RACER is just used to check the correctness of the ontologies; JEOPS is used to reason on them;

- I-X [Tate, 2001]: supports the collaborative planning process and it is the principal source of information to our system. Depending on the plan that I-X generates, our framework must capture the semantic of this plan and display it according to such a semantic and other current domain features (e.g., device restrictions). The <I-N-C-A> [Tate, 2000] ontology is used to represent such plans;

- J2ME [Sun Microsystems, 2003]: the Java 2 Micro Edition language, which is the Java version aimed at mobile devices, is used for the deployment of visualisation suites for handsets. Visualisation suites are independent modules of implementation of visualisation modalities.

7.5.4 Knowledge Representation

Our framework represents the knowledge in different formats. Here we summarise such formats:

- OWL [W3 Consortium, 2005e]: the Web Ontology Language (OWL) is used to represent the models/ontologies of the system. Any instance of a particular domain must be specified using the OWL format so that the system can load it;
- First Order Logic (FOL) [Russel and Norvig, 2003]: FOL is used as the primary language for the specification of visualisation rules;
- JEOPS Representation: as detailed later, rules in FOL are mapped to the JEOPS representation, so that they can be converted into a class that represents the knowledge base. This base receives facts in the form of Java objects, which represents instances of the models;
- Multi-Modal Visualisation: this is the final representation of a planning information in a information visualisation format.

7.5.5 Mappings Involved

This section summarises the mappings between different forms of information representation that occurs in our framework. Such mappings are:

- OWL -> Java Objects: according to the framework proposal, all the instances of its models are saved in a OWL format. At runtime, such instances are loaded by the system and translated to Java objects so that they can be inserted into the knowledge base;

- FOL -> JEOPS Rules Syntax: rules specified in FOL must be mapped to the JEOPS representation, so that the JEOPS engine can infer facts using such rules;

- Java Objects + JEOPS Rules Syntax -> Multi-Modal Visualisation: both, facts represented as Java objects and rules represented in Jeops syntax act together to transform an abstract representation of a plan in perceptive information to users, in the most possible and appropriate modality of visualisation.

The ideas introduced here given a better initial impression of the whole framework. Chapter 8 returns to such ideas, giving a more detailed explanation about them.

Chapter 8

Framework - The Reasoning Mechanism Services

The second part of the framework concerns the reasoning mechanism that will work upon and extends the ontologies discussed in Chapter 7. The ontologies were developed with the objective of facilitating reasoning. In this chapter is discussed how the reasoning mechanism takes place on the framework, to provide ways to reason and give outputs regarding information visualisation in the contextual environment of collaborative planning.

This chapter is organised as follows. Section 8.1 discusses the reasons why the framework reasoning was designed as a Production System. Section 8.2 introduces the rules purpose, how they are classified and what kind of decisions/reasoning such rules provide. Section 8.3 presents the architecture of the reasoning mechanism. Section 8.4 discusses the formal design and specification of the rules. Finally, Section 8.5 gives some details regarding the inference engineer, while Section 8.6 stresses important details about the reasoning mechanisms.

8.1 Information Visualisation Reasoning as a Production System

One of the main aims of our approach is the search for generality. In fact, a solution for information visualisation in planning systems does not intend to be dependent on current technologies, or attached to a specific planning approach, or based solely on existing devices, etc. In this way, a solution based on knowledge representation and reasoning can satisfy these requirements, since it provides ways to structure the problem and its semantics, independently of specific features of scenarios. Thus such solution were considered during the investigation and development of the framework proposed.

Knowledge representation provides a symbolic representation of a problem and its automatic manipulation via reasoning programs. Therefore, the base and focus of the framework is on knowledge, instead of, for example, purely functional aspects. In this way, the *Framework Part I* (Chapter 7) was concerned with what it is necessary to know about a domain, or what the relevant knowledge is (knowledge representation). In addition, the *Framework Part II*, discussed in this chapter, is dedicated to the investigation on how to make the knowledge available through computational mechanisms (reasoning). Note that an approach based on knowledge is able to represent the domain knowledge related to the information visualisation decisions, while also acting as a specialist in this domain.

The field of knowledge representation and reasoning is always concerned with the trade off between representation expressiveness and computational effectiveness. The ideal situation would be to use a representation as rich as possible and also be able to reason as effectively as possible. However the trade off between these two aspects forces an interplay between representation and reasoning.

Considerations regarding knowledge representation were discussed in Chapter 7. As for the reasoning part, the approach adopted in this thesis is that decisions associated with information visualisation are taken via a *Production System*, where a set of rules represents the knowledge about which is the most appropriate form of visualisation in a specific context. This context is specified in a pre-defined way via the ontologies described in the last chapter.

A *Production System* [Russel and Norvig, 2003] is a specific class of rule-based systems, which consists of a set of IF-THEN rules (implications), a set of facts, and some interpreter controlling the application of the rules, given the facts. The left hand side contains information about certain facts and objects, which must be true in order for the rule to potentially execute. Any rules whose left hand sides match are placed on an agenda. Then, when one of the rules on the agenda is picked, its right hand side (implication) is executed in the agenda. The agenda is then updated, and a new rule is picked to execute. This continues until there are no more rules on the agenda.

Mycin [Davis et al., 1977] is a traditional and good example of a rule-based system. Its job was to diagnose and recommend treatment for certain blood infections. An

English version of one of Mycin's rules could be described as:

IF the infection is primary-bacteremia
AND the site of the culture is one of the sterile sites
AND the suspected portal of entry is the gastrointestinal tract
THEN there is suggestive evidence (0.7) that the infection is bacteroid.

This rule clearly shows the use of the IF-THEN structure, which we intend to use in our system. However the strategy of *Mycin* is to first ask the user a number of more or less preset questions that are always required and which allow the system to rule out totally unlikely diagnoses. In our case the preset is defined by a scenario via instances of the ontology set defined previously. In this way, the use of reasoning on rules is more similar to modern examples of rule-based systems, such as in computer games [Champandard, 2003], which use rules to accomplish movement behaviors, weapon selection or tactical reasoning depending on parameters such as current spacial position and the situation of the game.

However, differently from logic programming languages, the consequence of implications in production systems are implemented as action recommendations rather than simply logical conclusions. Actions include *insertions* and *deletions* from the knowledge base as well as input and output. Thus, the rules in this thesis deal with two special functions in the implications consequences:

- Assert(f), which means, add the fact f to the knowledge base;
- Remove(f), which means, delete the fact f from the knowledge base;

These kind of operations are important for our methodology because, for example, the system can add new options of visualisation at runtime, which are actually recommendations. Thus a new fact, the visualisation option, is verified against other rules that decide if it holds.

Another important difference of production systems is their control structure. While most of the logic programming languages, such as Prolog, are backward chaining; production systems generally operate in a forward-chaining mode. Note that the backward chaining approaches search for a constructive proof that establishes some substitution that satisfies a query. This is not natural in our domains where we do not have queries. Instead, we have a knowledge base with a set of fact, described via the ontologies, and inference rules are applied to this knowledge base, yielding new assertions.

A last and important feature of production systems is the possible existence of *conflict resolution* mechanisms that decide which action to take when more than one is recommended. For example, a conflict resolution strategy could be the preference for rules that refer to recently created facts. Later on in this chapter we will detail our strategy for conflict resolution.

We could consider some alternatives to using product systems in our approach. For example, *Case-Base Reasoning* (CBR) [Aamodt and Plaza, 1994], *Supervised Learning* [Caruana and Niculescu, 2006] and *Fuzzy Logic* [Za68]. Using CBR, our system would have a knowledge base containing previous specific situations as opposed to visualisation rules. If the current situation is similar to some of the previous situations, the system could try to make decisions based on decisions that were taken in such previous situations. A major problem with using a CBR system, though, is determining how one situation is ”similar” to another. We found no previous works to determine similarities between visualisation needs, so it is difficult to implement the CBR approach in our domain.

If we had used the supervised learning approach, we could have found a visualisation function from training data, which consists of pairs of input objects and desired outputs. Then, the visualisation mechanism, using this function, could predict the value of visualistion outputs for any input scenario after having seen a number of visualistion training examples. To achieve that, the system should also generalize from the presented data to unseen situations in a reasonable way. Supervised learning still suffers from a similar problem as CBR, in that it is not very useful if there is no pre-existing information, which in this case would be visualisation training examples, from which to draw ideas. Furthermore, and CBR as supervised learning approaches are not easily extensible because they need additional information (situations and training examples respectively) to ensure an appropriate performance of their reasoning mechanisms.

We could have used Fuzzy logic if we were interested in handling uncertainty during the visualisation decision process. Systems that can handle uncertainty eliminate the restriction of a simply true or false by adding the proportion of something being true or false. If a system can determine the degree of truth in a given situation, it is more likely to be able to respond with more detailed feedback, such as how it came to its conclusion, which aspects of its decision are true or false, and so forth, rather than simply giving a true or false answer. However we do not see a significant level of uncertainty in the visualisation domain to justify the use of Fuzzy logic. For example, the rules related to devices does not give space to uncertainty because they conclude if some device is able (true) or is not able (false) to support a specific visualisation.

8.2 Reasoning on Visualisation Ontologies - The Decision Process

The idea proposed for the visualisation reasoning process is to allow that it creates the most appropriate interface in accordance with the scenario and knowledge specified via the ontologies. The first step to understanding this process is to associate groups of rules with the information codified for each ontology. Then, the reasoning can deal with group of rules, giving priority to some of them.

Based on this introduction, the reasoning process works on four principal groups of rules, which we call *scenario rules*:

- The **device-restriction rules** analyse the device specification to decide which categories of visualisation are allowed, thus filtering the rules that can infer a suitable option(s);

- The **planning information-restriction rules** consider mainly, but not only, the type of planning information being visualised to take decisions about convenient methods;

- The **agent-restriction rules** analyse the agent requirements regarding its needs and preferences for the task that is being executed. Based on that, suitable methods of information visualisation are proposed; and

- The **environment-restriction rules** decide which are the appropriate forms of visualisation based on awareness and characteristics of the environment and restrictions that it can impose.

An alternative thought what *device-restriction* rules is to consider that these rules are restrictions on the use of components that can be used during the creation of the interface. In this way, rules restrict the components domain, remembering the principles of constraint satisfaction problems. The function of the rules associated with the devices ontology is exactly that. If a device is not able to support some component or category of visualisation, then such component or category must be removed from the reasoning process.

This strong notion of restriction associated with devices justifies the development of rules that use its ontology as a first filter. For example, if the specification of a device says that it only supports components of the text category, all the other rules that can infer any other category can be eliminated.

Considering the other extreme, we could imagine a powerful device that can deliver at the same time more than one category of visualisation. In this case the reasoner can create an interface with several components to display the same information in different ways. Part of the agent ontology, which specifies preferences, can be used during this process to lead the creation of the interface when there are several options available.

While the first group of rules is used to decide which visualisation categories can be used, a second group accounts for reasoning about the content of the visualisation. The plan specification (via the planning ontology) is the most important source of information for this reasoning. The process must verify the description of the plan to identify elements, such as temporal constraints, which have a most suitable way to be delivered.

The rules defining the reasoning about *planning information-restriction* are based on two main aspects: first, different types of planning information require different approaches for visualisation; second, the same information can be viewed in different styles. For instance, when visualising information regarding to world states, such as the wind direction in a collaborative planning operation. That piece of information can be visualised, for example, either in a more sophisticated graphical way or in a very simple textual description.

Rules regarding *agent-restriction* are concerned with agents profiling and preferences. Agents profiling would characterise agents in terms of the role it is playing in the planning process and in the agents organisation. For instance, if the agent is performing a task on the move that requires the use of hands, it might be more appropriate to formulate a solution for information visualisation with the help of sound alerts. In addition, human agents can also set their own preferences on how to visualise information.

Finally, the rules related to the environment ontology apply conditions and requirements regarding environment awareness, with more emphasis in location-based information.

The result of the reasoning mechanism is represented by what these group of rules together are going to decide. That is, a strategy for information visualisation in the context of a collaborative planning environment. The next section explains the process of decision making used by the reasoning mechanism architecture.

8.3 Reasoning Mechanism Architecture

This section presents the reasoning mechanism architecture, which is illustrated in Figure 8.1. As explained in the previous chapter (Chapter 7), instances of the four ontologies' classes (Device, Agent, Planning Information and Environment) define a scenario. Given a scenario definition, the reasoning mechanism infers a suitable information visualisation modality, expressed semantically by classes of the Information Visualisation ontology. For that, the reasoning mechanism occurs in two main phases. In the first phase the *Scenario Rules* are applied. As a result, several suggestions of suitable information visualisation are proposed as output. In a second phase, optional *Filtering Rules* can be applied to choose only one modality of information visualisation among the proposed output set.

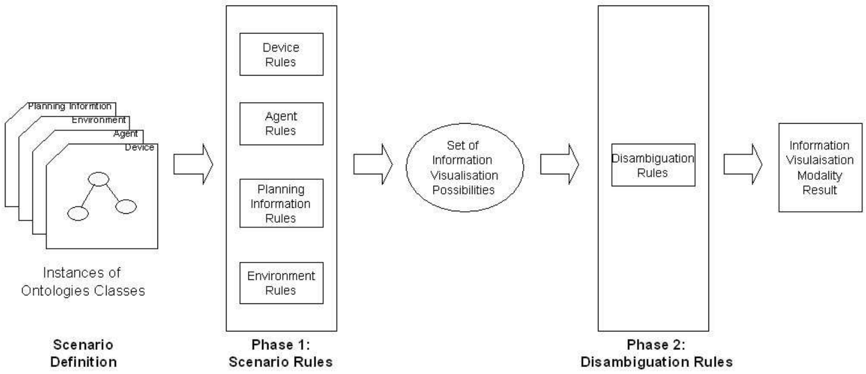


Figure 8.1: Reasoning mechanism architecture.

The *Scenario Rules* are rules related to the four ontologies that semantically define a scenario. As was introduced in the previous section they are:

- Device-restriction Rules;

- Planning Information-restriction Rules;
- Agent-restriction Rules; and
- Environment-restriction Rules.

The *Filtering Rules* are rules designed to bypass ambiguity in cases where the *Scenario Rules* leads to more then one option of information visualisation. These rules are explained in Appendix C. Next section (Section 8.4) presents the design and formal specification of each of these groups of rules. The group of rules are presented with examples that would permit the reader to identify the purpose of each group. However, the complete list of rules was subtracted from this chapter, but are available at Appendix B for further consultation and reference.

Lets consider now the specification of a scenario, which is used in the remaining of this chapter to discuss the design and formal specification of the rules. To semantically specify a scenario, we need to instantiate classes of the four ontologies. For that end, lets first recall the I-Kobe domain, introduced in Chapter 3, which is a knowledge-based model inspired on the Kobe Earthquake, and also a domain application of the broader I-Rescue project.

A list of agents modelled in the I-Kobe domain were presented on Table 3.1. For our scenario definition, to discuss the reasoning mechanism, we consider an agent of the type Fire Brigade that works on a tactical level of the hierarchy. The ability of this agent is to extinguish fire and, for that, it has a set of characteristics associated with its skills, such as water capacity, length of ladder, etc. In addition, this agent also has preferences, including the ones regarding information visualisation. This agent profile is illustrated on Table 8.1.

Agent Type	Fire Brigade
Hierarchical Level	Tactical
Ability	Extinguish fire
Water Capacity	2,000l
Ladder Length	20m
Visualisation Preferences	No

Table 8.1: Scenario definition: agent in the I-Kobe scenario.

Lets consider the Fire Brigade agent is on the move collaborating in the planning process, making use of a mobile device. This mobile device would be a PDA, model

Palm Tungsten E2, with display resolution 320 x 320, processor Intel XScale 200 MHz ARM, 32 MB non-volatile flash memory. Furthermore it has as specific characteristics a virtual keyboard and 5-way navigator for providing ways to access information and navigation through that. Also, the device would be Java enable, what semantically means that it is capable of running all Java 2 Micro Edition (J2ME) capabilities, plus additional APIs and/or libraries installed.

Device Type	Mobile Device
Model	Palm Tungsten E2
Resolution	320 x 320
Number of Colours	65,536
Processor	200 MHz ARM
Memory	32 MB
Java Enabled	Yes
Configuration	CLDC
Libraries Available	Network Algorithms Library
Hardware Navigation Capabilities	5-way navigation

Table 8.2: Scenario definition: device in the I-Kobe scenario.

Regarding the planning information being requested and manipulated by the agent, in the context of the collaborative process of planning, lets consider that the Fire Brigade agent has the plan activity of extinguishing the fire in Kobe Tower. The refinement for this activity is the set of the following activities: go to refill place, refill water tank, go to Kobe Tower, extinguish fire. There are also some world constraints defined for the plan and its resources, for instance: the water tank has a full condition; the fire brigade ladder has height measure of 20m; the status of Sokoba Road is clear; and the status of Nikuso Avenue is also clear.

Activity	Extinguish Fire in Kobe Tower
Refinement	Yes
World Constraints	Yes

Table 8.3: Scenario definition: planning information in the I-Kobe scenario.

To conclude our scenario example, it needs definitions regarding the environment ontology and its location-based information type. For instance, it is relevant for the activity being performed by the Fire Brigade agent the position of refill place and the position of the Kobe Tower.

Latitude refill place	58.98
Longitude refill place	31.9876
Latitude Kobe Tower	59.08
Longitude Kobe Tower	30.9987

Table 8.4: Scenario Definition: Environment in the I-Kobe scenario.

Based on this scenario definition, the reasoning mechanism can take place, as it is going to be illustrated through examples in the remaining of this chapter. It is important to remember that the agents in this scenario are structured in a hierarchical organisation, as discussed in Chapter 3. In this way, we do not have a global central component, such as the facilitator agent in OAA. Hierarchies support the scalability of the system, because they have local central agents that coordinate only the parts of agents that are immediately under their level of decision. Each agent of this hierarchy has its own visualisation reasoning mechanism, so that the number of agents does not have an influence on this mechanism.

8.4 Rules Design and Formal Specification

This section specifies and explains the rules used in our system. To that end, the rules are divided in classes according to their main functions. Note that the rules described here only represent a subset of the rules that could be needed in a real system. One of the advantages of a rule-based system, however, is that it can be easily extended. This extension only needs to consider the classical problem of conflict that can appear between the current and new set of rules. Next subsections describe each group and their rules.

8.4.1 Device-restriction Rules

The rules of this group make statements and reason about the devices, based on the ontology/vocabulary specified on Chapter 7, to generate as a result suitable ways of information visualisation.

The rules of this group are divided into the following categories:

- Basics: rules that define ways of information visualisation based on basic features of devices;
- Java Technology Semantic Based: in this category are included rules for Java enabled devices, assuming the standard functionalities of the mobile platform J2ME;
- Display x Sound x Navigation Semantic Based: in this class are included rules that explore the specific features that mobile devices have for the usage of displaying, sound and navigation; and,
- Advanced and New Technologies Semantic Based: in this category are included rules to deal with more sophisticated ways of information visualisation.

This set of rules is associated with the constraints of each class of devices, considering the attributes and definitions given for these classes. The main function of this set is to remove the visualisation modalities that are definitively impossible to be used due to physical restrictions of the device in use.

Initally, all the modalities are added to the base, together with the device instance that is going to be used. In this way, the device rules must indicate which modalities are supported for it. The following rule, for example, codifies the conditions that a device needs to have to support the 3D (virtual reality) modality. Such conditions are, for example, physic constraints (video data transference rate) and existence of support library (OpenGL or DirectX).

> ∀d,m DEVICE(d) ∧ MODALITY(m) ∧ isModality(m,3D) ∧ hasMinimumVideoDataTransfer(d,m) ∧ hasOpenGlOrDirectXLibrary(d)
> ⇒ enabled(m)

Predicates DEVICE(x) and MODALITY(y) mean that the instances x and y are from Device and Modality classes respectively. If this rule holds, its consequence is the assertion of a new fact to the basis saying that the modality “m” is now enabled to be used. In this way, only the enabled modalities will be used for the remainder rules during the reasoning process.

8.4.2 Planning Information-restriction Rules

This set of rules is mainly designed to verify the content of a plan and return a list of tuples linking a plan, or parts of it, to visualisation modalities. Thus, the plan ontology is the principal source of facts which these rules act on.

According to our approach, every plan *p* is composed by elements *e*, according to the <I-N-C-A> ontology. When a plan *p* is created, its elements are added to the knowledge base as facts, which will validate one or more rules during the reasoning process. For example, consider the following rule:

$$\forall \text{p,e,m } \text{Plan(p)} \wedge \text{ElementOf(e,p)} \wedge ((\text{m = Textual}) \vee (\text{m = Tabular}) \vee (\text{m = NLP}) \vee (\text{m = Sonore})) \Rightarrow \text{DisplayEnabled(e,m)}$$

According to this rule, for every instance of the plan class, the information related to any of the plan element of this instance can be delivered via a textual, tabular, NLP or sonore representation. In other words, we are saying that these modalities are appropriate to deliver any kind of plan information represented by <I-N-C-A>. This rule, in particular, only consider the kind of plan element (issue, activity, constraint or annotation) to generate a conclusion. However, other rules need to analyse specific features of each plan element. For example, consider the rule in follow:

$$\forall \text{p,a } \text{ActivityOf(a,p)} \wedge \text{hasRefinement(a)} \Rightarrow \text{DisplayEnabled(p,a,Tree)}$$

According to this rule, for every activity of a plan, if this activity has a refinement, then it can be visualised via a tree representation. This means, the use of a tree representation for activities is only appropriate if there is an associated refinement because refinements create a hierarchical structure for activities. Thus, this rule has a special function, called ”hasRefinement”, that accounts for analysing the internal structure of the activities to discover if there is one or more refinements.

A similar case is presented by the rule in follow, this time to the constraint element:

$$\forall \text{p,m,c } \text{PLAN(p)} \wedge \text{CONSTRAINTS(c)} \wedge \text{isModality(m,3D)} \wedge \text{enabled(m)} \wedge \text{hasTridimensionalDescription(c)} \Rightarrow \text{visualisation(c,p,m)}$$

This rule links the constraint set of a plan to a 3D visualisation modality. According to the rule, a 3D (virtual reality) modality is justified if the set of constraints of a plan has a tridimensional description component. Again, we need a function that implements the meaning of “hasTridimensionalDescription”.

8.4.3 Agent-restriction Rules

This set of rules analyses the agent requirements regarding its needs and preferences for the task that is being executed. This work considers two optional concepts that are related to the planning process, however they are not essential for it: organisational structure of the group and their own description of agents. The first concept is important because it places each agent in the planning process, highlighting its function. The second shows the preferences and mental state of each agent, stressing what they can do or intend to do during the planning from its own perspective.

The rule in follow, for example, says that if there are two options to visualise a same planning element, the agent preference could be used to decide for one of them.

$$\forall p,e,c\ visualisation(e,p,m_1) \wedge visualisation(e,p,m_2) \wedge \neg(m_1{=}m_2) \wedge agentPreference(e,m_1) \Rightarrow retract(visualisation(e,p,m_2))$$

8.4.4 Environment-restriction Rules

This set of rules decides which are the appropriate forms of visualisation based mainly on the characteristics of the environment and restrictions that it can impose. Such rules are used together with the planning information rules to configure appropriates manners to deliver the planning information.

8.5 Reasoning Example in Kobe Scenario

Let us use the scenario defined in the last section to exemplify the use of our approach. In an initial stage, the process has a knowledge base representing all the facts (plan, device, agent and environment) about the domain (Fig 8.2a). The first step is to apply the device rules that account for discovering the possible modalities of visualisation that the device, in this case the Palm Tungsten E2, supports. Example of rules are:

- For every instance of the device class, if this instance has Java capabilities and if this instance has a CLDC configuration, then it has available the MIDP profile;
- For every instance of the device class, if it has the MIDP profile, then it supports special Java applications;
- For every instance of the device class, if it supports special Java applications and it has a tree algorithm library, then this instance supports the tree visualisation modality.

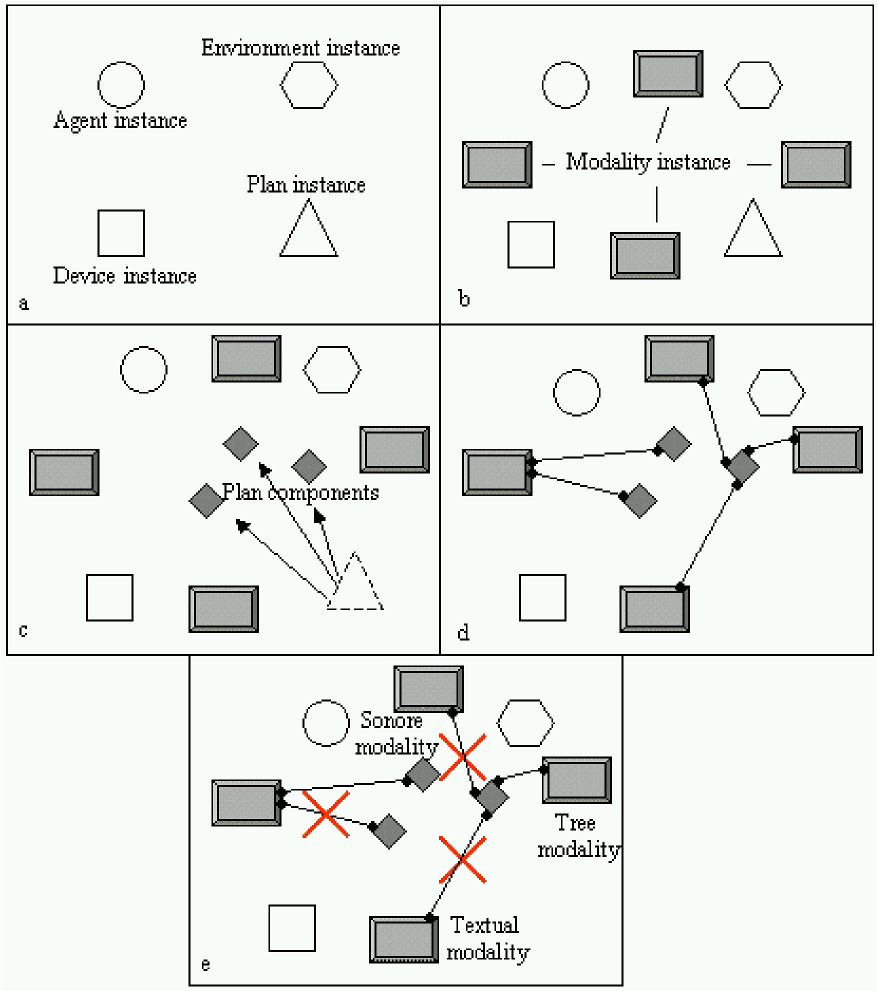


Figure 8.2: Knowledge base in five different moments during the reasoning process.

By applying these rules and others of the same category, all the possible modalities are enabled into the knowledge base (Fig 8.2b). The subset of rules listed above, for example, enables the tree visualisation modality in particular.

The next step is to split up the plan into different parts that could require a different visualisation option (Figure 8.2c). For that end, the process applies the plan rules, such as the example that follows:

- For every instance of the plan class, if such an instance has a set of activities, then every activity of this set is an activity of the plan.

This rule is a way to verify if the plan has a set of activities. After identifying the plan components, such as the activities, the planning rules, together with the environment rules, associate these components with appropriate modalities of visualisation

(Figure 8.2d). The rule in follow is an example in this direction:

- For every activity of a plan, if this activity has a refinement, then it can be visualised via a tree representation.

This rule specifies that a tree is an option to visualise activities and their refinements. The problem is there are other options to visualise activities, such as the tabular and sonore modalities, which are the default visualisation alternatives. At this point the process uses filtering rules, which are in fact a kind of conflict resolution strategy that gives priority to some kinds of rules. In this example, a conflict resolution strategy could be defined to say that if there is the option of delivering the activities information via the tree modality, then this one must be used (Figure 8.2e).

In order, the knowledge base must implement more general strategies such as “try first *complex* modalities”. In this way, the tree modality will take advantage in relation to text or sonore modalities. The other option is to show the same information in different ways. Generally this option is not very useful in real missions, however the base of rules could easily be extended to support rules that deal with such an approach.

8.6 The Inference Engines

In this project we have used two different inference engines: the *RACER OWL Reasoner* [Haarslev and Moller, 2003] and JEOPS (Java Embedded Object Production System) [Figueira and Ramalho, 2000]. The use of these two engines is detailed in the next subsections.

8.6.1 RACER

RACER provides an integrated environment with Protege so that the ontology set, which was specified via Protege, can be directly used by this engine. While this integrated environment of editing and tests allows an easy evaluation of rules and the integrity of the ontologies, RACER does not provide an easy way to integrate its engine with other components developed in Java, which is the language used for the I-X architecture development and for our prototype. As RACER provides an OWL reasoner and inference server, Java applications can use the network classes to access this server via the TCP/IP protocol. This could be an option if RACER was a free open source rather than a proprietary code.

Based on these comments, we decided to use RACER only in a first stage as a quick validator of ontologies. One of the main services offered by RACER is to test whether or not one class is a subclass of another class. By performing such tests on all of the classes in an ontology, it is possible to compute the inferred ontology class hierarchy. Another standard service that is offered by a reasoner like RACER is consistency checking. Based on the description (conditions) of a class, the reasoner can check whether or not it is possible for the class to have any instances. A class is deemed to be inconsistent if it cannot possibly have any instances.

One option for the core reasoning of our framework would be using RACER and SWRL (Semantic Web Rule Language) [W3 Consortium, 2004b] combined. One important consideration is that despite the fact that RACER has support for applying SWRL rules to instances specified in an OWL ontology in a server-based environment; however by the time of having to take technological decision during our investigation, RACER SWRL engine was still being extended to deal with OWL datatypes. A first version including this aspect was available in RACER 1.9.

For that reason we investigated other solutions and opted for using the JEOPS inference engine.

Nevertheless, the use of RACER and SWRL as an inference engine should be explored in future works, mainly regarding the Semantic Web integration of the framework. This solution would make the approach even more standard-based and following the Semantic Web concepts and trends.

8.6.2 JEOPS

After the logical consistency checking of the ontologies via RACER, we have used JEOPS to reason about the visualisation rules[1]. JEOPS is a Java-based inference engine whose principles are similar to RACER. Both approaches offers a forward chained engine that applies the rules until no new information is added. A forward chaining system starts with initial facts and keeps using the rules to draw new conclusions (or take certain actions) given those facts. Consequently, forward chaining systems are primarily data-driven, what is in accordance with our idea of reasoning. In other words, we have all the data about a specific scenario and the goal is to find a better visualisation mode for that scenario.

In JEOPS the initial rules are mapped to the JEOPS format showed in follow:

[1]Note that RACER can also play this role. RACER automatically maps initial rules to a Semantic Web Rule Language (SWRL), integrating such rules into the OWL ontology

```
ruleexample
    declarations
        space for variables declaration
    conditions
        space for rule conditions
    actions
        space for rule effects
```

Then a rule base file, which contains rules in this format, is pre-compiled into a java file that implements the inference engine, according to the rules of the original file. The advantages of this methodology is that there is not limitation of java types and expressions and every Java piece of code can be used in the rule action part. However this could be also seen as a disadvantage because implementations using JEOPS become less unconstrained.

All the information codified via the ontology set is inserted into the knowledge base as instances of objects. Then, if there are objects in this knowledge base for the declarations, and all the expressions in the conditions evaluate to true when the variables are instantiated with those objects, then the body of the actions field will be executed. For example, considering the following rule:

```
rule3 dEnable {
    declarations
        Device d;
        Modality m;
    conditions
        !m.enabled()
        m.isType("Virtual Reality")
        d.videoDataTransferRate() > m.getMinVideoTransfer()
        d.hasLibrary("OpenGl");
    actions
        m.setEnabled(true);
        modified(m)
}
```

This first order production rule in JEOPS syntax stresses the relation between the FOL rules and the object-oriented notation. Apart some details, like the use of *modified* as a special function to inform object updates to the knowledge base, the mapping process between FOL based rules and JEOPS is very natural. Together with *modified*, other special functions used in the next chapter are:

- *retract*, to remove objects from the knowledge base;
- *assert*, to add a new object into the knowledge base.

Note that based-rule systems do not guarantee by themselves that a given set of rules will terminate or achieve a conclusion. The rules specifiers account for ensuring this aspect so that they must have the ability to create a suitable set. In this way, the set creation is more related to the specifiers skills than a matter of engineering.

The scenarios, evaluations and conclusions carried out using JEOPS are detailed in the next chapter.

8.7 Analysing the Reasoning Process

This section analyses two important aspects of the reasoning process: the match algorithm and the knowledge base definition. This last aspect focuses on the rules ordering and its effects on the visualisation mechanism.

8.7.1 The Match Algorithm

Our application uses the *Rete Algorithm* [Forgy, 1982] to deal with the problem of matching facts to rules. This algorithm is implemented by building a network of nodes, each of which representing one or more tests found in a rule. Facts that are being added to or removed from the knowledge base are processed by this network of nodes. At the bottom of the network are nodes representing individual rules. When a set of facts filters all the way down to the bottom of the network, it has passed all the tests of a particular rule and this set becomes an activation. In other words, this set is able to active a rule so that its implications can be executed.

The principal idea of this algorithm is to improve the speed of forward-chained rule systems by limiting the effort required to recompute the activation set after a rule is fired. For that, it considers two observations:

- Temporal Redundancy: the firing of a rule usually changes only a few facts, and only a few rules are affected by each of those changes;

- Structural Similarity: the same pattern often appears in the left-hand side of more than one rule.

The disadvantage of this algorithm is its high memory space requirements. However its broad use in several known production systems (e.g., Jess, CLIPS, etc.) suggests that the gain in performance compensates for this problem. In fact our prototype is not complex enough to present problems in terms of memory space. Independently of the application complexity, it is important to understand the reasons for this drawback to avoid problems in a possible real version of our proposal.

Within the network itself there are broadly two kinds of nodes: *one-input* and *two-input* nodes. One-input nodes perform tests on individual facts, while two-input nodes perform tests across facts and perform the grouping function.

The two-input nodes have to integrate facts from two different inputs that we call left and right inputs. Any facts that reach the top of a two-input node could potentially contribute to an activation. The two input nodes therefore must remember all facts that are presented to them, and attempt to group facts arriving on their left inputs with facts arriving on their right inputs to make up complete activation sets. Therefore a two-input node has a left memory and a right memory and this is the point where the disadvantage of this approach appears.

The example in follow clarifies the practical use of the Rete Algorithm. First, consider the two rules bellow:

$\forall$p,c ConstraintOf(c,p) $\wedge$ Type(c,Resource) $\Rightarrow$ DisplayEnabled(p,c,Tabular)

$\forall$p,c ConstraintOf(c,p) $\wedge$ Type(c,Resource) $\wedge$ Has2dPosition(Object(c))
$\Rightarrow$ DisplayEnabled(p,c,Map)

Such rules might be compiled into the network illustrated in Figure 8.3. Note that there are shared nodes in this network. This means that patterns of two different rules (e.g.,ConstraintOf) are represented by the same node because they are similar. This is one of the methods to simplify the match process. Each node of this network has a memory that keeps the values that turn this node true. For example, the node Type(c,Resource) will keep all the constraints ”c” whose type is Resource. Nodes like that are the one-input nodes and they perform tests on individual facts, while the two-input nodes (nodes marked +) perform tests across facts and perform the grouping function.

As the nodes keep all the information about past test results, only new facts are tested against only the rules to which they are related. For example, a new constraint will be tested only against rules that have a constraint as parameter.

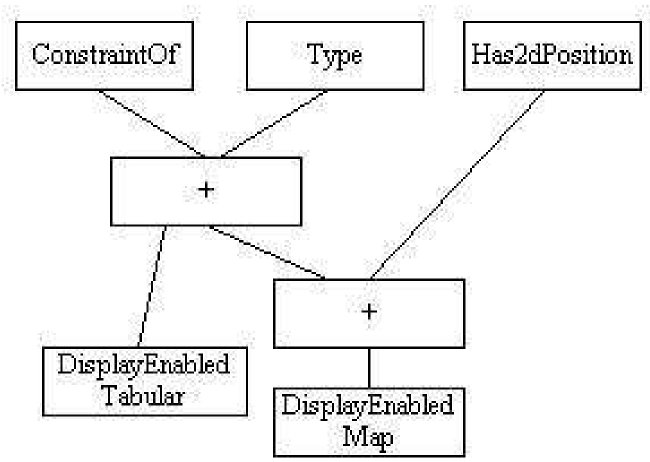


Figure 8.3: Simple example of network representation for two visualisation rules.

8.7.2 The Knowledge Base Definition

For a better understanding of the knowledge base definition, lets first summarise the reasoning process. The mechanism works in cycles, so that in each cycle the engine verifies all the rules, according to the Rete algorithm, and all the valid rules are considered candidates to be executed. This set of valid rules composes the conflict set (or activation set). Then some resolution strategy is applied to the conflict set so that an order is imposed to the fireing of such rules. After that, new cycles are sequentially started until no rule is fired.

In our case, the order in which the rules are fired is important because the rules of device, plan, environment and agents must be applied in this sequence[2]. To ensure this sequence, we are using the *Priority Strategy*, which gives priority to rules that are first declared in the rules base.

In practical terms the ordering of the rules does not guarantee that the mechanism reach a solution. However the ordering has influence on the number of cycles. For example, the device group rules are the unique group that enables modalities instances to be used by other rules. This is the reason that it should be defined sooner. If it is defined at the end of the base, then the mechanism needs at least one more cycle to make available such modalities.

Apart the conflict resolution to rules, there is also the conflict resolution for visualisation results. In this case we are using the filtering rules, as discussed before. This

[2]The agent rule group can be used to set preferences on visualisation modalities, so that they are applied as a kind of exclusive-output filter in the cycle.

set acts as a final filter in cases where only one visualisation modality is required. In this way, its correct position is in the final of the rules base. A clear example of problem is when such rules are defined before the agent preference rules. In this case the preferences are not likely to be considered because the filtering rule will fist perform a filter in the possible modalities, commonly removing the preferred modality.

Part IV

Application Scenarios, Validations and Conclusions

Chapter 9

A Practical Application

This chapter shows how the framework proposed in this thesis can be used in a practical application. The application is based on a disaster relief operation where several agents are carrying out tasks in a collaborative environment. A disaster relief domain is a good example for our demonstration because it involves agents using several kinds of devices and dealing with different parts of a plan. In this way, Section 9.1 introduces the application domain and the agents involved. Section 9.2 details the system setup, showing the use of the ontologies descriptions by the reasoning mechanism. Finally, Section 9.3 discusses a running section of the system.

9.1 Characterising Domain and Agents

The domain used in this demonstration is based on an urban disaster relief scenario, such as the The *Great Hanshin* or Kobe Earthquake. Such an event is an example of how natural disaster have tragic effects in urban areas. On Tuesday, January 17th 1995, at 5.46 a.m. (local time), an earthquake of magnitude 7.2 on the Richter Scale struck the Kobe region of south-central Japan. This region is the second most populated and industrialised area after Tokyo, with a total population of about 10 million people. The ground shook for only about 20 seconds, but in that short time over 5,000 people died, over 300,000 people became homeless and damage worth an estimated 100 billion was caused to roads, houses, factories and infrastructure (gas, electric, water, sewerage, phone cables, etc).

We can classify the agents that are performing in this environment into three representative classes: (1) Central command and control agents, (2) Local command and control agents and (3) Execution agents. Note that inside each of these classes can coexist several command and control levels. However the basic three levels idea is still the same.

The important point of this classification is that agents in each group are likely to use different devices, depending on the role that they are performing in the organisation and their location. While central command and control agents commonly have powerful resources available, execution agents will have limited type devices that do not disrupt their mobility and action. Local command and control agents could have an intermediary kind of device between powerful and limited ones.

Another important point in this discussion is that the planning process, performed for each of these classes, is also different. The next three tables (Table 9.1, 9.2 and 9.3, based on [Siebra, 2006]), describe this difference.

The central command and control level (Table 9.1) accounts for developing plans at a high level of abstraction, or "what-to-do" plans. In other words, the level specifies what must be done, but it does not give details about how something must be done. In this way, the principal tasks are related to analysis, directions and comparison of courses of actions.

Feature	Description
Input	Generally a complex and abstract task
Output	Requests for the performance/filling of "what-to-do" plans
Time	Long-term goals
Influence	The entire coalition is affected by its decisions
Knowledge	Global, diversified and non-technical
Processes	Problem analysis, definition of directions and priorities

Table 9.1: Central command and control agents.

Considering a disaster relief domain, this level could be represented by the *Search and Rescue Command Centre* (SRCC). Just after an earthquake, the SRCC receives the tasks of rescuing injured civilians and limiting the damage to the city. Analysing the problem, the SRCC decides to divide the city into regions and set priorities for each of them (some regions can be more critical than others because they have a higher probability of having buried civilians, historic value such as museums and monuments, or present risks of increasing the catastrophe such as deposits of fuel and explosives).

The SRCC can also analyse global information, such as speed and direction of wind to predict the fire behaviour and generate tasks to avoid future causalities. Possible outcomes of its deliberative process are: avoid the fire spread to region *x*, look for buried civilians in buildings of region *y*, keep unblocked the road *z* (because it is an important path to access resources), and so on. Note that such outcomes say what must be done without references on how they must be done. Furthermore they are long term goals, which can affect the entire coalition.

The local command and control level (Table 9.2) could be composed of local units such as fire stations and hospitals. When such components receive subgoals from the strategic level, they start by checking the necessary conditions and options to reach the subgoals, according to their available resources. In this way, operational components are taking decisions at a different level because they are thinking about how the activities can be carried out.

Feature	Description
Input	What-to-do plans and possible restrictions on their performance
Output	Requests for the performance of specific tasks
Time	Mid-term goals
Influence	One or more sub-coalitions are affected by their decisions
Knowledge	More specialised, mainly on the operation environment and resources
Processes	Synthesis of plans, resource allocation, load balancing, etc.

Table 9.2: Local command and control agents.

Each local unit has the function of employing its subordinates to attain specific goals through the design, organisation, integration and conduct of sub-operations. For that, each unit has its own skills and abilities so that its knowledge is more specialised in the field in which it is operating. This level also pays significant attention to the resource/time relation. This means an efficient and balanced use of resources. Thus, processes such as automatic task allocation and load balancing are very useful.

The level of execution (Table 9.3) is where the execution of operations actually takes place. For this reason the degree of knowledge of tactical components is very specialised within the domain which they are operating, and their decisions are generally taken on sets of atomic activities. As the components are performing inside a dynamic and unpredictedable environment, their reactive capabilities and speed of response are very important so that the use of pre-defined procedures could be an useful alternative. The output of this level is a set of atomic activities that are commonly executed by the own components.

Feature	Description
Input	Specific tasks and possible restrictions on their performance
Output	Primitive operations (atomic activities)
Time	Short-term goals
Influence	Decisions should not have influences on other levels
Knowledge	Very specialised
Processes	Pathfinder, patrolling, reactive procedures, knowledge sharing, etc.

Table 9.3: Execution agents.

The execution level, in a disaster relief operation, could be composed of fire brigades, paramedics and police forces for example. For the performance of their tasks, these components could need specific intelligent processes such as a pathfinder, which looks for best routes to specific destinations, or patrolling mechanisms to trace routes that efficiently cover search areas. The tactical level is also the principal source of new information to the coalition because its components are in fact moving through the environment. In this way they are more propitious to discover changes and new facts that must be shared among their partners.

From this discussion the diversity of information and planning processes in a disaster relief domain is clear. However, as discussed before, this is not an exclusive feature of this domain, so that several collaborative planning domains present this same diversity.

9.2 The Framework Setup

Consider that each member of a disaster relief team has an assistant agent[1] *a* running in a device *d*, dealing with a subplan *p* in an environment *e*. To run our framework we must have:

- A description for *a*, according to agent ontology, which must be loaded to *d*;
- A description for *d*, according to device ontology, which must be acquired from the own device;

[1]Note that in case of a computational entity, such as a robot, the agent is the own entity.

- A description for *p*, according to plan ontology, which is produced by a planning process running inside the device;

- A description for *e*, according to environment ontology, which must be loaded to *d* before the start of the operation.

The first step of the visualisation mechanism is to transform all these descriptions into objects to be inserted into a knowledge base. For example, the device is one object and the attributes of such an object represent the features of the device. This process is illustrated in follow (Figure 9.1).

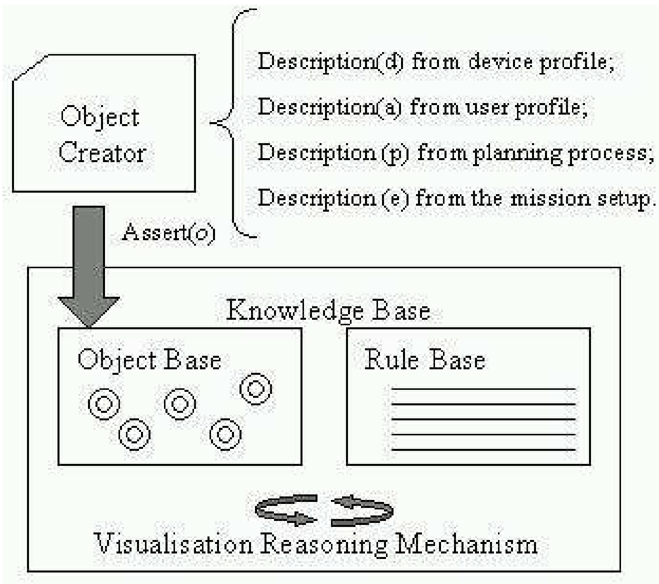


Figure 9.1: The internal architecture of the visualisation reasoning mechanism.

Before starting the visualisation reasoning, a specific component of the architecture, the *Object Creator*, obtains all the descriptions from different sources and creates the instances that compose the object base. In the same way, all the modalities must also be loaded into the object base in the form of objects.

In brief, the following steps must be performed during the practical use of our framework:

1. Load the objects representing all the visualisation modalities to the object base;

2. Load the object representing the device profiling to the object base;

3. Load the object representing the user profiling to the object base;

4. Load the object representing the operation environment, if there is this information, to the object base;

5. Load the object representing the plan, which is being manipulated by the agent, to the object base;

6. Run the rule base on these objects.

The outcome of the reasoning is one or more mappings from visualisation modalities to the plan or, more commonly, parts of the plan. This sequence was used during the setup of our experiment, which is detailed in the next section.

9.3 Running the Application

This application uses a subset of the rules defined in Appendix B, which are used together with each of the following scenarios (Table 9.4):

Scenario	Agent	Device	Filtering rules
1	Operation commander	C2 Room	no
2	Fire Station	Personal Computer	no
3	Fire Brigade	Mobile Device 1	no
4	Fire Brigade	Mobile Device 1	no
5	Operation commander	C2 Room	yes
6	Fire Station	Personal Computer	yes
7	Fire Brigade	Mobile Device 1	yes
8	Fire Brigade	Mobile Device 2	yes

Table 9.4: Definition of scenarios in terms of agents, devices and employment of filtering rules.

Agents are characterised in the last section, while the plan specifications used in these scenarios are available via the web[2]. Lets then, define each of the devices:

- C2 Room: command and control room[3] with processing power of 2 parallel processors of 6.0GHz, 2GB RAM memory and four 40” (1920x1080) LCD Flat Panels. Hard memory of 300GB, containing all libraries;

[2]http://www.aiai.ed.ac.uk/project/ix/project/lino/plans.zip

[3]Examples of C2 rooms can be seen in http://www.control-centers.com/pages/AlliedSignal/index .html and http://www.evansonline.com/products/consoles/response/

- Personal Computer: a Pentium 4 Processor 3.0 GHz, with 512MB memory and a 20" (1280x1024) LCD Flat Panel as display. Contains following visualisation libraries: GUI, DirectX and Map;

- Mobile Device 1: Motorola V980 Handheld with processing power of 200 MHz, 2MB memory, a 30x20 display, CLDC configuration and Java enabled. It does not contains any special library;

- Mobile Device 2: Palm Intel XScale 416 MHz, de 4GB memory, display 60x50 TFT, CLDC configuration and Java enabled. Contains special libraries to manipulate tree and network representations.

Each of the scenarios is an experiment and all of them use the same instances of visualisation modalities: textual, tabular, sonore, graphic, network, tree, spatial, virtual reality (3D) and natural language. After running the experiments, the system returns the options for each kind of plan element in accordance with the rules.

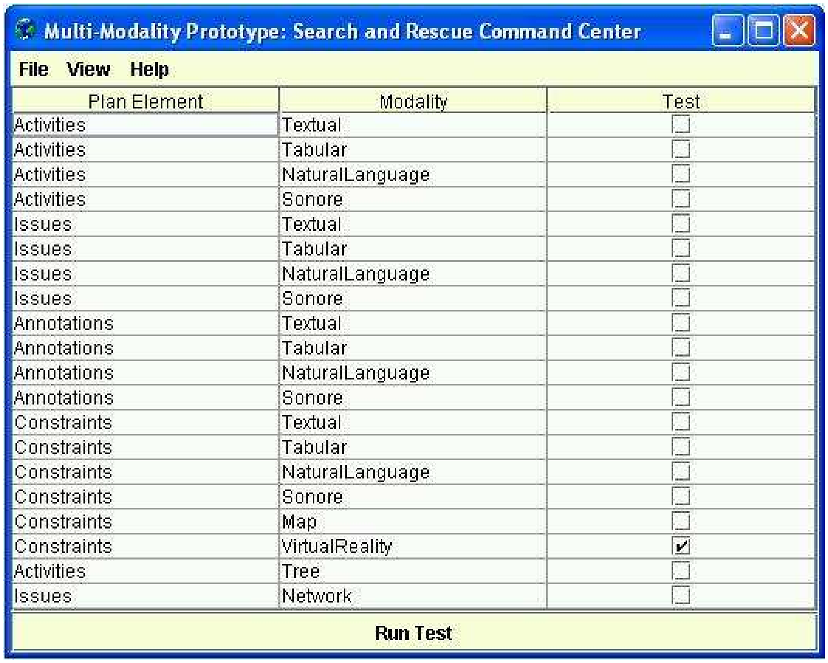
All the figures in following section show indications of visualisation modalities, returned by the system, to plan elements. Figure 9.2 shows the results to scenario 1. As the visualisation rooms are very well equipped in terms of hardware and software, they enable any kind of planning visualisation. So we can see several visualisation options as follows.

Figure 9.3 shows a smaller set of visualisation options (from now on we are no longer considering issues and annotations for simplification reasons). There are two motives for that. First the device resources are more limited, mainly in terms of libraries. Second the user has set a visualisation preference constraints so that if this option is available (in this case the map modality), only this option is returned.

Figures 9.4 and 9.5 show results of experiments that use the same agent profile running in different devices. The second device (Figure 9.5) is more powerful, however it returns less options because it provides the kind of visualisation modality that was set by its user (Map modality). Note that, if the system infers that the first device (Figure 9.4) does not support the map modality, then the agent preferences cannot be applied and all other possible options are returned.

In the majority of planning systems, one kind of visualisation is enough for each plan element. Thus, cases like the one represented in Scenario 1 (Figure 9.2) must be refined.

The refinement process is carried out via filtering rules, as previously explained.



Multi-Modality Prototype: Search and Rescue Command Center

File View Help

Plan Element	Modality	Test
Activities	Textual	☐
Activities	Tabular	☐
Activities	NaturalLanguage	☐
Activities	Sonore	☐
Issues	Textual	☐
Issues	Tabular	☐
Issues	NaturalLanguage	☐
Issues	Sonore	☐
Annotations	Textual	☐
Annotations	Tabular	☐
Annotations	NaturalLanguage	☐
Annotations	Sonore	☐
Constraints	Textual	☐
Constraints	Tabular	☐
Constraints	NaturalLanguage	☐
Constraints	Sonore	☐
Constraints	Map	☐
Constraints	VirtualReality	☑
Activities	Tree	☐
Issues	Network	☐

Run Test

Figure 9.2: Visualisation results to Scenario 1.

Note that user preferences can be considered one kind of filtering rules, however in situations that they cannot be applied, then the system must offer some filtering strategy.

The strategy used here to exemplify the idea of filtering is simple. If there is one or more special structure modalities, one of that modalities is aleatory chosen. Otherwise, the system tries one of the complex structure modalities. If both options fail, then one simple structure modality is used. In brief, the idea is to try more specialised modalities before the simple ones.

Note that for this kind of reasoning, the system needs to understand the hierarchical relation between the classes (Figure 7.2). For example, it needs to know that if the *Tree modality* is part of the *N_Dimensional* set and the *N_Dimensional* set is part of the *Complex_Structure* set, then the *Tree modality* is also part of the *Complex_Structure* set. Appendix 3 details the rules that use this strategy.

Adding the set of filtering rules (Appendix 3) to the rule base, we have the following results (Figure 9.6). Note that the system returns only one visualisation modality for each category, according to the new set of rules.

One implementation feature that was not discussed yet is the ability to test the resultant visualisation[4]. The right columns of the tables (Figures 2 to 6) present check

[4] This feature is not implemented to all modalities of the model yet.

Figure 9.3: Visualisation results to Scenario 2.

Figure 9.4: Visualisation results to Scenario 3.

boxes that can be checked and a run test method can be started to the associated visualisation. The I-X architecture gives support to integration of different ways of visualisation via the *Object-viewing whiteboards* approach. This feature enables that Java classes, which implement visualisation modalities, can be added at runtime to the planning architecture. In this way, the classes can access the required information from the architecture to create specific visualisations, and also the alterations carried out via the interfaces can be reported to the architecture. To make this test more realistic, we are setting the size of the test window in accordance with the device's display (information from the device ontology).

From this practical demonstration we can conclude that the use of rules-based reasoning is an appropriate approach to deal with this domain. First because this kind of

Figure 9.5: Visualisation results to Scenario 4.

reasoning is a declarative way to codify the know-how of the domain. Second it is very simple to change the rules if we want to represent new knowledge or a new process. In other words, this approach eases the system extension and maintenance.

9.4 Demo Screens

In the previous sections we saw how launch the application to run the framework. In this section we will present the interfaces for information visualisation of the prototype demo.

Taking for example the visualisation results of the framework, shown in Figure 9.6, if the user decides to see a specific type of modality he/she should click on it and then press the button to run the demo, that will show the respective interface.

Following, the screen shots of some of the modalities developed in the prototype system are presented.

Figure 9.7 to 9.12 show respectively: the Textual modality, the Tablular modality, the Tree modality, the Network modality, the Map modality, and the Virtual Reality Modality interface.

It is interesting to note that the same information (for example, building locations) can be displayed in different modalities, according to which is more suitable in different situations.

Figure 9.6: Visualisation results to Scenario 5 to 8 respectively.

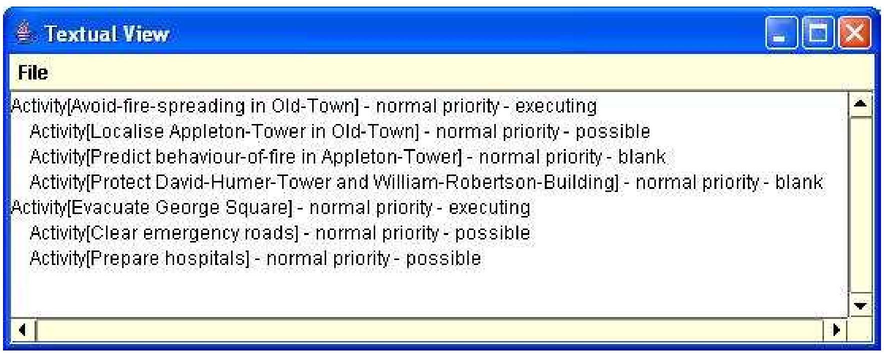


Figure 9.7: Textual modality interface.

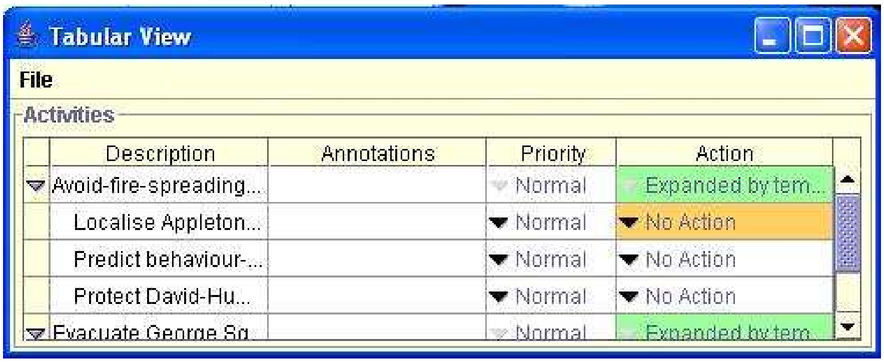


Figure 9.8: Tabular modality interface.

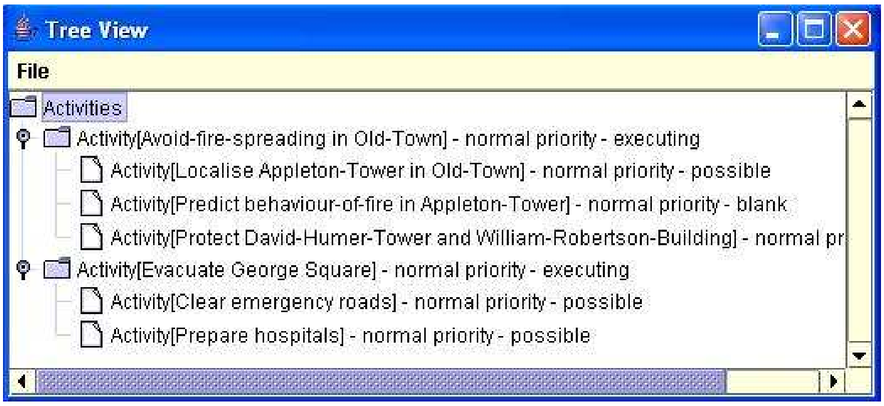


Figure 9.9: Tree modality interface.

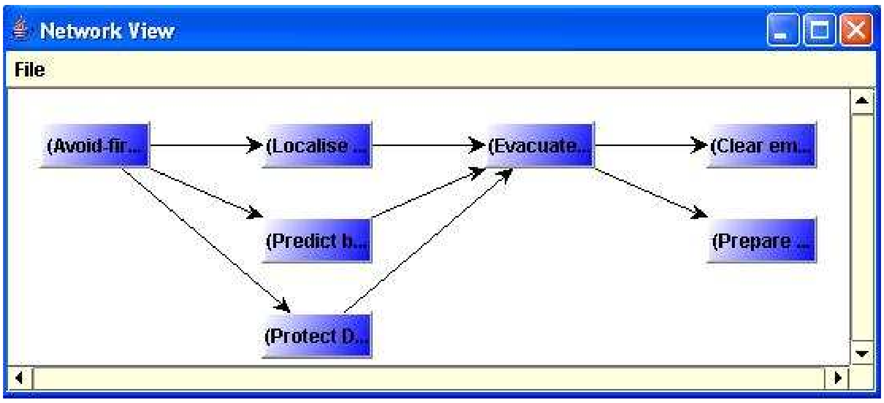


Figure 9.10: Network modality interface.

Figure 9.11: Map modality interface.

Figure 9.12: Virtual Reality modality interface.

Chapter 10

Conclusion

This chapter contains conclusive discussions about our work. In brief, the work consists of a framework of semantic based support for visualisation in a context of complex collaborative planning environments. It is intended to be a generic and to enable the organisation and modelling of planning domain from the visualisation perspective, giving tailored support for information visualisation.

The remainder of this chapter is organised as follows: Section 10.1 empirically evaluates the framework, using some results from the experiments in Chapter 9, according to a set of requirements or criteria set. Section 2 stresses the main contributions of this work. Finally, Section 10.3 lists future works that could be carried out from this work.

10.1 An Empirical Evaluation

This section discusses an empirical evaluation of our framework, which uses results derived from experiments of Chapter 9 and related observations. For that we follow the methodology of first defining the scope of the framework. Then, we list the set of requirements that the framework tries to cover, showing if they are fulfilled.

The idea of our framework is to consider any kind of collaborative planning domain, which can be defined via a planning representation language. Because we are using a specific representation, the <I-N-C-A> ontology, as a basis for our planning model, we can say that the scope of our framework is delimited by the coverage of <I-N-C-A> in representing planning domains.

Based on this assumption, we need to analyse the coverage of <I-N-C-A> itself. The proposal of <I-N-C-A> is to be a general ontology for the representation of plans.

In this way, it is based on general objects (e.g., activities, constraints, etc.) rather than concepts coupled to particular domains. To cover a broad set of planning domains, <I-N-C-A> objects are specified in a very open way. The content of constraints, for example, is defined by a list of *parameter* elements, where *parameter* is an open kind of element that will be defined according to the constraint to be created.

While this kind of definition provides enough freedom to create several kinds of constraints, the semantic of new constraints cannot be directly used by the reasoning mechanisms. In this way, it is important that a more refined definition of constraints is given, via the definition of types such as world state, temporal or resource constraints. Then, the reasoning process can correctly use the elements of these definitions. For example, the previous definition of temporal constraints allows that a set of this kind of constraints to be analysed to create or choose a customised form of visualisation delivery to this specific set.

A conclusion to this discussion is that the scope of our framework is restricted to all kind of domains that can be specified via the version of the <I-N-C-A> ontology presented in Chapter 7. However, this itself is very broad. Note that expansions in its representation will not have an impact on our framework. However such expansions will not aggregate value to the visualisation reasoning process, just because the framework will not recognise them. Considering this scope, we can evaluate our framework according to five requirements: coverage, extensibility, soundness, completeness and quality.

The evaluation of coverage tries to investigate if the framework covers all possible scenarios, or if there is any type of problem/event that such a framework does not cover and why. As discussed before, the scenarios represent domains of collaborative planning, such as the Search and Rescue instance discussed in Chapter 9. This domain has been used because it is a complex real world area of concern, involving several agents and types of devices. In this way, its employment was useful because we could verify that the models were able to represent the significant domain features from the point of view of the visualisation needs. For example, we have used very different visualisation devices to see how they could be modelled. In fact, independently of the device type, all of them have a subset of features that can be specified by the framework models. Examples of these features are display size, sound support or processing power.

The evaluation of extensibility examines if the framework can easily be modified to consider expansions in the models. This requirement is closely related to coverage.

The current framework has a specific coverage given by the models and rules. If the framework has a good extensibility, then it is also easy to update the coverage of the framework. The design of our framework has mainly considered this requirement via the use of a rule base to keep knowledge about the visualisations. As discussed before, a rule base can easily be extended and maintained. Also, the categorisation of these rules and reasoning, proposed in this thesis (Section 7.4), enables a better understanding of the process and, consequently, supports the insertion or modification of new rules. We can feel these features during our experiments when the filtering rules are used. This new set of rules has a significant impact on the results, however its design and integration into the framework is simple and direct.

The evaluation of soundness examines if the framework behaves correctly and as expected. An advantage of this framework is that the models can be previously tested via RACER, which provides a way to test for inconsistencies and structural errors in the models. Related to the inference process and rules, we have used eight instances of test the scenarios (Chapter 9) to verify the correctness of the rules. Using simple observation of the outcomes, we could verify if such outcomes are actually appropriate and follow the ideas codified via the rules specifications. Note however, that this is not an exhaustive kind of test, so that the use of multiple variations may bring some unexpected result.

The evaluation of completeness examines if the framework covers all of the necessary concepts and functionalities. At its current stage, our framework is not meeting this requirement. There exist concepts associated with the environment and agents that are not being explored in their entirety. As discussed during the thesis, these concepts can have an influence on the visualisation process, apart from the fact that they are not fundamental for such a process. In fact several concepts can be added to the models, as well as rules to augment the quality of reasoning.

The evaluation of quality examines how well the framework covers/supports the problem domain. In other words, it examines the quality of results. We have noticed that quality is closely associated with the definition of rules. Note that the soundness of the framework does not imply that the results are the most appropriate for a given scenario. During the development of the experiments, we have considered the search for quality when we try to match the best form of visalisation to each plan element. For example, the match of temporal constraints elements to the temporal modality of visualisation. In this case, the rules are mainly in charge for the results quality. However an interesting situation noticed in our experiments is when users have visualisation preferences. Because such preferences have priority in the inference process, the become the users responsibilty. Maybe it sounds inappropriate that the framework generates a final result that is different of the agent preference, because we are claiming that the framework always looks for the best visualisation modality. This could indicate that agents do not know the real capability of their devices, or they do not feel comfortable with such a specific modality. This last case shows that quality is a subjective parameter so that the same result could be attested as high quality for some user and not so good for another.

10.2 Contributions

This section lists the main contributions of our framework, discussing each of them in details.

10.2.1 Generality

The framework was designed to be a general approach, in opposition to what was designed to date in AI planning systems. What we have in the past in visualisation in AI planning systems are specific fixed solutions, limited by many reasons:

- Dependant on the style of internal planning representation;
- Dependant on the planning output;
- Not flexible to different requirements: old approaches only provide a pre-defined way to visualise information despite the requirement differences. These requirements can be of different natures, such as, user requirements, devices for visualisation requirements, type of planning information requirements, etc. For instance, considering user requirements, those can vary depending on: hierarchical role of the user agent in the collaborative process of planning (strategic, operational, or tactical); and
- Limited to current technologies for visualising information.

On the other hand, the framework proposed in this thesis consists of a general framework. It attacks the problem from the conceptualisation of it, defining a high level abstract model composed by the following components or building blocks:

- Conceptual Models: that permit the definition of a scenario and its requirements;

- Reasoning Mechanism: based on the conceptual definition of scenarios. According to requirements and restrictions that they impose, the framework provides a reasoning mechanism that has as output a selection of visualisation modalities suitable for the scenario;

- Visualisation Modalities Conceptual Models: these conceptual models specify different modalities of information visualisation. These models, at the same time, work as a conceptual specification for visualisation modalities and are used as output for the reasoning mechanism. In such outputs, the visualisation modalities can come individually (if the filtering rules are applied), or in a set (when the filtering rules are not applied). In this last case a set of visualisation modalities are presented, leaving the user with the option to choose between them. The application or not of the filtering rules in the reasoning mechanism process would be utilised in the framework according to the requirements. For example, an user agent working at the strategic hierarchical level would be interested in having several modalities of visualisation output. This would give the agent the possibility to go through the options to analyse the information from different perspectives and also being able to delegate and give advice about information visualisation to subordinated and/or peer agents. On the other hand, the use of filtering rules would more suit the tactical agents that are, for instance, executing the plan. In such cases giving one solution for information visualisation (the most suitable one for the scenario according to the requirements), would speed the process of analysing information for the agents working at this level.

- Visualisation Suites: These constitute solution blocks for each information visualisation modality. Using the application based on the framework, the user can run a scenario for a given agent taking part in collaborative process of planning, and the application will return one or more options of information visualisation, depending on the use or not of the filtering rules. The user can then, choose one of the options by ticking it, and then run the visualisation suite referent to the information visualisation modality that was ticked. The visualisation suite tries to simulate as precise as possible the information visualisation according to the scenario requirements. For instance, preserving the display size of an hypothetical mobile device, etc.

The first part of Figure 10.1 (on the left) illustrates how the information visualisation was approached to date in other solutions. The second part (on the right) shows how the framework presented in this thesis proposes to solve the problem using a general solution to it.

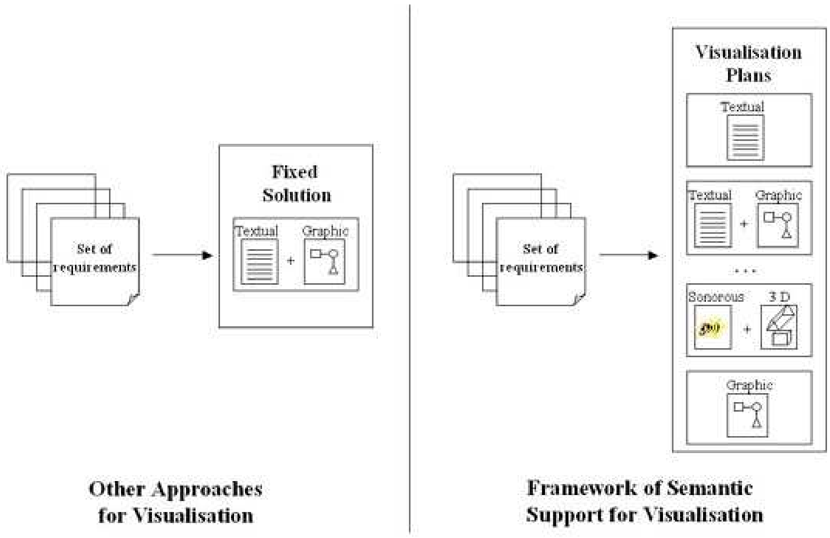


Figure 10.1: Approaches for information visualisation in AI planning systems: other approaches proposed to date (left) and approach proposed in this thesis (right).

The framework proposed consists of a high level abstract model for information visulisation in collaborative AI planning. The approach was not designed to be limited to current technologies of intelligent planning, information visualisation, or mobile computing, etc. Instead it is open and extensible to new technologies through conceptual formalisation. Therefore it consists of a general approach.

To illustrate the generality of our approach, let us consider what could happen if we try to use it in another collaborative planning domain, such as the Mars rover mission. First, several visualisation devices in space missions could be very different from the ones used in a disaster relief operation. However the features (e.g., screen size, processing power, etc.) of space devices tend to be the same than any other device. In this way, rather than extending or changing the ontology, we only need to create new instances of this ontology to represent the space devices. The same idea can be used to the agent and environment ontologies. A fireman and an astronaut are different agents, but they have the same set of properties that characterises them. This is also valid to the Mars terrain or a disaster scenario. These objects are only instances of ontologies, so that they share the attributes and rules specified for their respective ontologies. This aspect is very important to the maintenance of rules. The rules of our approach are intended to manipulate features and relations specified by the ontologies, rather than on specific devices, or agents or environments. Thus, in the majority of the cases, our approach only requires the creation of new instances for each new domain.

All these aspects are also valid to the planning ontology. A disaster relief plan and a space mission plan are certainly very different. However they are based on the same concepts such as activities, issues, constraints (temporal, world-state, etc.) and so on. On the other hand, the generality related to the planning ontology is restricted to plans that are specified via <I-N-C-A>. Planners that generate plans in a different language, such as PDDL or STRIPS, will not be able to use this framework. Note, however, that ever in these cases, only the planning ontology and planning rules should be modified. Another option could be to map the plan representation in use to the <I-N-C-A> syntax. However, depending on the representation in use, this may not be a very practical process.

10.2.2 Extensibility

For the development of our approach, the framework consisted of a semantic model of a subset of concepts involving elements of scenario and information visualisation definitions, these definitions were useful to validate our framework. However, the framework was designed to be a general conceptual model. Thus, the framework can be easily extended to incorporate new cases. These can include new scenario specifications via, for example, the addition of new devices for visualisation; and/or new and advanced modalities of information visualisation. For instance, it could be a pen-based (pen gesture and inking) [Li et al., 2005], [Hinckley et al., 2004] or tactile [Cholewiak and Beede, 2005] modalities of information visualisation. The framework supports this extension due mainly to the approach of using semantic modelling and a rule based reasoning mechanism, as already discussed during the thesis.

The methodology for extending the framework would need to follow a number of simple steps that we describe bellow. For instance, lets discuss the case for extending it for a new information visualisation modality, for example, pen-gesture. The methodology necessary for extending the framework for a new modality like this is defined by two steps:

- Adding the semantic conceptual definition for the modality of information visualisation (Information Visualisation ontology), and;

- Add the requirements of this new modality in terms of the restriction rules of the decision process, regarding device, agent, planning information and environment restriction rules.

Note that modifications would not be necessary for the current framework regarding its main modules, which are:

- Scenario semantic definition, and;

- Reasoning Mechanism approach using JEOPS as a production system.

In addition, modifications are also allowed regarding scenario definitions. For example, the inclusion of new devices, new agents, etc. This stops our framework being limited to current technologies, but allows it to be general and open to new possibilities.

10.2.3 Enhancement of the Use of Knowledge-Based Planning

The framework presented in this thesis has enhanced the use of knowledge-based planning in other areas, not restricting it to the core problems of intelligent planning. It is an attempt as a step ahead to a broader use of knowledge-based planning applied to the area of information visualisation in the context of collaborative planning.

It has been argued in the literature that there is a need for a broader use of knowledge-based planning based on the ideas of a knowledge enrichment, required in AI planning. However, as far as we are aware, it has only been investigated under the light of core problems of planning. Our claim is that this vision should be even more augmented in other aspects of planning, and we highlight, for instance, the information visualisation area.

The enhancement of knowledge-based planning permitted by our framework makes contributions in the following aspects:

- It is a first attempt to use a knowledge representation approach applied to the problem of information visualisation in collaborative planning systems. As already discussed in this chapter, previous solutions for information visualisation in planning systems presented dedicated approaches which were not very flexible in different situations. The knowledge-based approach presented in this thesis creates opportunities for a new way of thinking and developing information visualisation in planning systems;

- The knowledge models developed permit modelling and reasoning about the problem from the information visualisation perspective. In our framework they were designed to work together giving support to the reasoning mechanism. The reasoning mechanism gives output solutions for information visualisation based on the knowledge bases of structured information. In addition, the models and the structured information they provide can also be used separately for other aims and tasks. For instance, the device's model/ontology, which contains detailed descriptions of mobile devices, can be used to other applications and problem domains.

10.2.4 Designed for Real World Applications of Collaborative Planning

A strong notion of our work is that it can in fact be used in real collaborative planning applications. The design of the planning ontology, for example, was based on a framework (I-X) that already has several implementations in different kinds of domains (military, search and rescue, etc.). However, other knowledge models were developed only because the information raised from this planning ontology is not enough to fit the requirements of real work applications regarding to a multi-modality visualisation.

In a real world scenario, the most likely situation is human and software agents collaborating to solve a planning problem. Human agents will have different roles in the planning process. While some will be in coordination task, others will be on the move. This information is not explicit in the planning ontology, but it can be important in defining a visualisation strategy. In the same way, other information about agents, such as roles, capabilities, preferences and authorities in the planning process are also important and they can all be represented in the agent ontology.

The environment ontology follows the same fundamental approach: to augment the information about the domain so that a better visualisation strategy can be applied. Despite the fact that this ontology was not extensively explored during our research, its employment is essential to represent information about the environment that can have influence on the visualisation. Note that real world applications can be designed for very diversified environments such as space, underwater, underground and hostile environments (e.g., battlefields) where the kind of domain can impose several information delivery constraints.

Finally, the device ontology allows the use of a planning application on a broad range of devices. It is almost impossible to assume that a collaborative planning application, designed to real world domains, is going to be used for only one kind of device. Thus, this ontology brings the required knowledge to adapt the planning information to one or other specific device.

In brief, the set of ontologies and their integration permits the expressiveness of several aspects related to real world applications in planning domains. We can say that the whole set of ontologies gives us the power of adaptation. In other words, planning information is adapted to be delivered in such a way that it becomes compatible and appropriate to a given situation. Note that we are arguing that these four groups of information (planning, device, agent and environment) are enough to represent all the required information to decide on a planning delivery strategy.

Related to the reasoning mechanism, the number of rules required by a real application can affect its performance, so that it is important, for example, to recognise and avoid irrelevant rules. This is not different in our approach and a large number of rules will possibly require techniques of optimization, as detailed in [Gupta et al., 1986, Zupan and Cheng, 1998], which can speed the reasoning process.

At last, it is important to stress that this level of adaptation/representation/reasoning is not found in any kind of planning application. In fact, several principles discussed here could be used for any kind of computational system that requires some form of adaptation in its process of visualisation delivery. However, as any original approach, several improvements are still needed, so that there are different opportunities to research directions from our current stage.

10.2.5 Tailored Information Visualisation Delivery Based on Knowledge Representation

The framework proposed in this thesis allows a tailored delivery and visualisation of planning information, according to features of a scenario. This is achieved through an original approach of knowledge representation that, despite similar ideas having been investigated in the information visualisation field, this line of research has never been applied to collaborative intelligent planning applications.

There are two levels of tailoring in our framework: a non-disambiguating mode and a disambiguating mode. The system can be configured to work in both modes. In the non-disambiguating mode, all possible tailored visualisation modalities suitable to a given scenario are presented to the user as suitable information visualisation modalities.

On the other hand, in the disambiguating mode, only one tailored option is presented to the user, being that option which the reasoning mechanism elects as the most suitable for the given situational scenario.

10.2.6 Independent Models Usage

The ontologies set permits organising and modelling the domain from the visualisation perspective in a contextual collaborative environment of intelligent planning. The framework puts the ontologies together to work for this purpose. However, each model has a contribution in itself, since they can be used separately for different domains and applications.

For instance, the *Devices Ontology* can be used for devices profiling in any other application. The approach presented is motivated by the need for semantic enhancement for mobile device profiling. This work brings several contributions to the area via a broader knowledge representation regarding many aspects.

First, it permits semantic improvements related to Java technology. This will allow reasoning considering about Java aspects (resources, API's, plug-ins, etc.), enabling the reasoning mechanism to propose tailored modalities of information visualisation regarding the knowledge aggregated via this ontology structure.

Second, the *Devices Ontology* is also providing semantic enhancement related to display, sound and navigation aspects of mobile devices, motivated by the fact that a wiser use of these resources can improve mobile devices usability.

This enhanced knowledge can be used in the context of our framework. However, that aggregated knowledge can potentially be used in numerous other applications that need to deal with devices profiling for example.

10.2.7 It has a Conceptual Model

The framework was built, first of all, based on a conceptual model that serves as a base for implementations. Some requirements and features existing in the conceptual elaboration are:

- Not attached to a specific and unique way of information visualisation;
- Not attached to current technology; and
- Not attached to current available devices.

10.2.8 Originality

In addition, it is important to highlight the originality aspect of this work. A semantic modelling approach has not yet been applied to information visualisation in intelligent planning applications as far as we are aware. The use of ontologies is becoming a trend in the information visualisation field, where an increasing number of works relating to this subject have appeared in recent international conferences on the topic. However, its use in an intelligent planning context has not yet been explored.

10.3 Future Works

We are witnessing a fast development of the Web heading towards the next generation Web, which may be more semantically structured. There is a need for new research and technology challenges that will permit the continued Web growth and access. Some new technologies are being explored to address these challenges, that will extend the capabilities of the Web. Our framework can fit these goals. It can be extended and be applied in the Semantic Web [W3 Consortium, 2005a], permitting the engineering of new ways to access/visualise the Web.

An interesting extension from this work is related to its great potential for application in the Semantic Web [W3 Consortium, 2005a] and also the Semantic Grid [Blythe et al., 2003]. This extension is possible first because our framework was developed according to Semantic Web and Semantic Grid concepts. Second because it was developed based on real industry and academy standards and trends.

Semantic Web is an international research initiative, in which the core goal is to make web content available for intelligent knowledge processing. It is a vision of an evolving version of the current web in which the Web is a universal medium for data, information, and knowledge exchange.

In brief, the Semantic Web is a vision, a set of design principles, collaborative working groups, and a variety of enabling technologies. It is built upon to main aspects: common formats for integration and combination of data drawn from diverse sources, and language for recording how the data relates to real world objects. The methods and tools developed and integrated for the purpose of the Semantic Web, often called Semantic Technologies, are generic and have a very large application potential.

The fact that our framework is based on real industry/academy standards permits and eases its extension, communication and interoperability with other systems and services, including web-based services, and application on the Semantic Web.

The compatibility with the Semantic Web is given both in the conceptual level (semantic modelling based on ontologies) and implementation level (code specification based on W3C [W3 Consortium, 2005d] standards (e.g., OWL [W3 Consortium, 2005e]).

Our approach could contribute with the Semantic Web/Grid in different directions. Following, we describe a few:

- Automatic update of the knowledge bases: The knowledge bases defined by the ontologies/models in our framework, for example, the devices ontology, can be automatically updated for the inclusion of new devices descriptions using intelligent agents and semantic technologies on the web.

- Reuse of models/ontologies: The five ontologies (Planning, Devices, Agents, Environment and Multi-modality Visualisation) developed for our framework can be reused in the context of the Semantic Web, permitting also the creation of new versions of meta-models.

- Customised information visualisation on the web: The framework can be extended to information visualisation in the context of the web, according to information in a standard definition, to provide web-based information visualisation customisation.

- Support to semantic-enabled software engineering: Semantic-enabled software engineering is the combination of Software Engineering and Semantic Technologies. Semantic Technologies includes: Ontologies, Ontologies Builder, Semantic Web Services, Semantic Web, Reasoning, and Reasoner Standardisation. Since the approach developed in this thesis is based on several of these technologies, it has potential for the integration with methodologies and practices of Software Engineering. For instance, Requirements Engineering done through Knowledge Acquisition, development of ontologies and their re-use through the whole software development process.

- Contribution to standardisation in environments of collaborative intelligent planning: This work draws on standards (OWL, <I-N-C-A> based on RDF) and novel techniques (semantic modelling and ontologies) trying to improve the lack of semantic rich descriptions of e.g. functionality and quality attributes; and intending to provide data interoperability, and automatic orchestration of components and services in the domain of collaborative planning.

Apart from the Semantic Web opportunity, we can list other directions of research:

- Extension of the models, so that they can mainly consider more features of agents, the environment and devices. This also implies an extension of the rule base, so that it also reasons on the new model classes and instances;

- Improvement of the evaluation tests, which must consider each of the requirements described in Section 10.1. Such tests should, for example, consider more than one planning domain to see the behaviour of the framework and to prove its generality;

- Practical implementation of visualisation tools that represent the visualisation modalities described by the model. These tools should be integrated to the I-X architecture and employed at runtime.

Note that the principal ideas of these work directions is to verify the framework in a more concrete way and also to turn it into a real planning tool, integrated to the I-X architecture.

Appendix A

Semantic Modelling

A.1 Ontologies

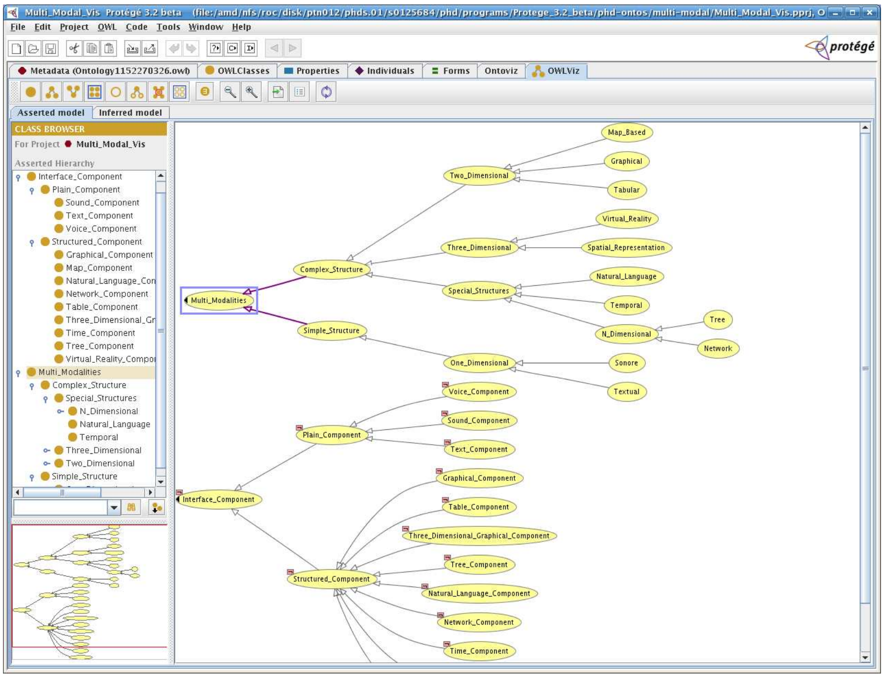


Figure A.1: Multi-Modality Visualisation ontology.

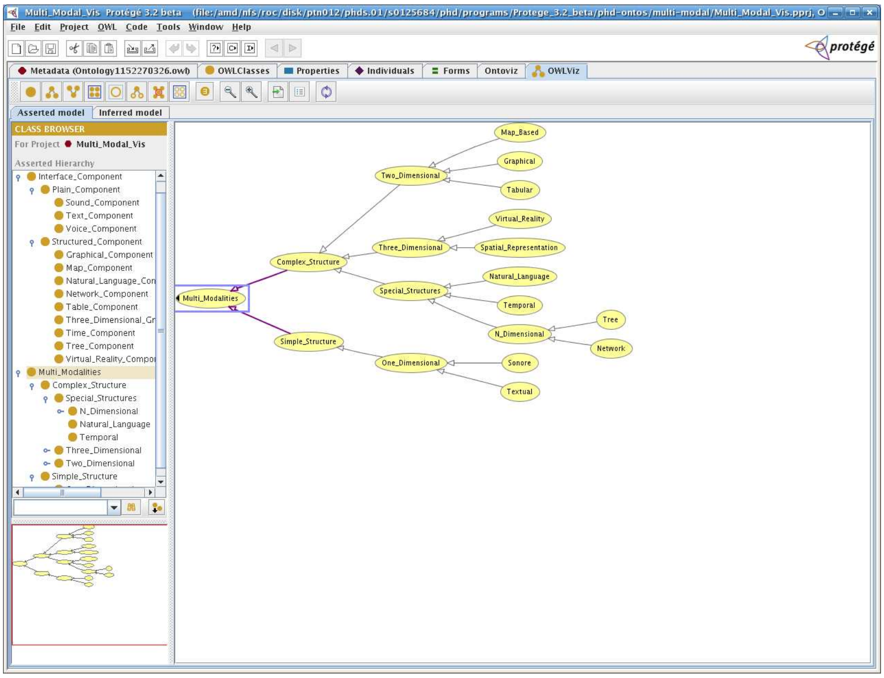


Figure A.2: Multi-Modality Visualisation ontology - Multi-Modalities focus.

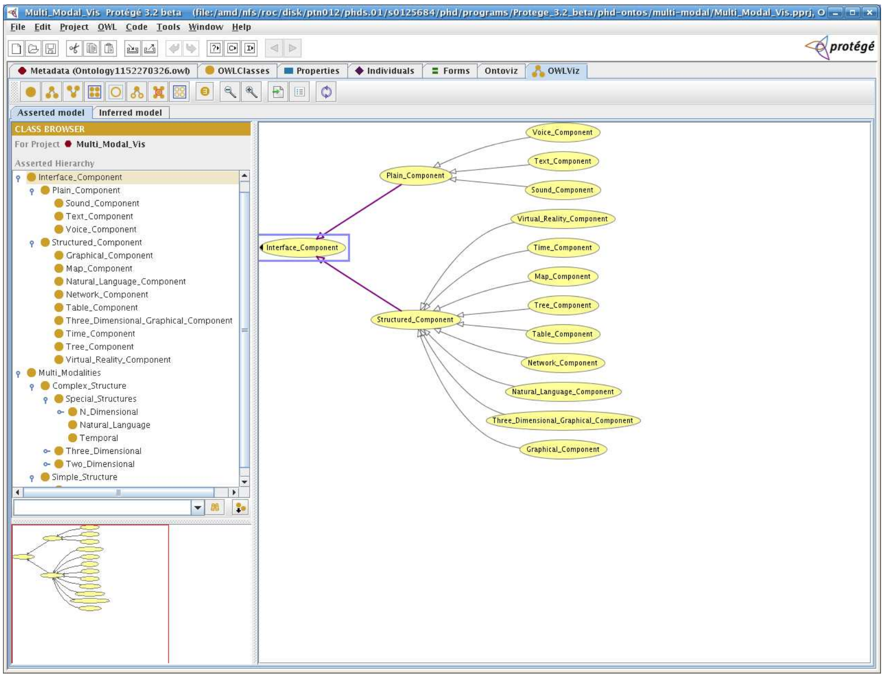


Figure A.3: Multi-Modality Visualisation ontology - Interface components focus.

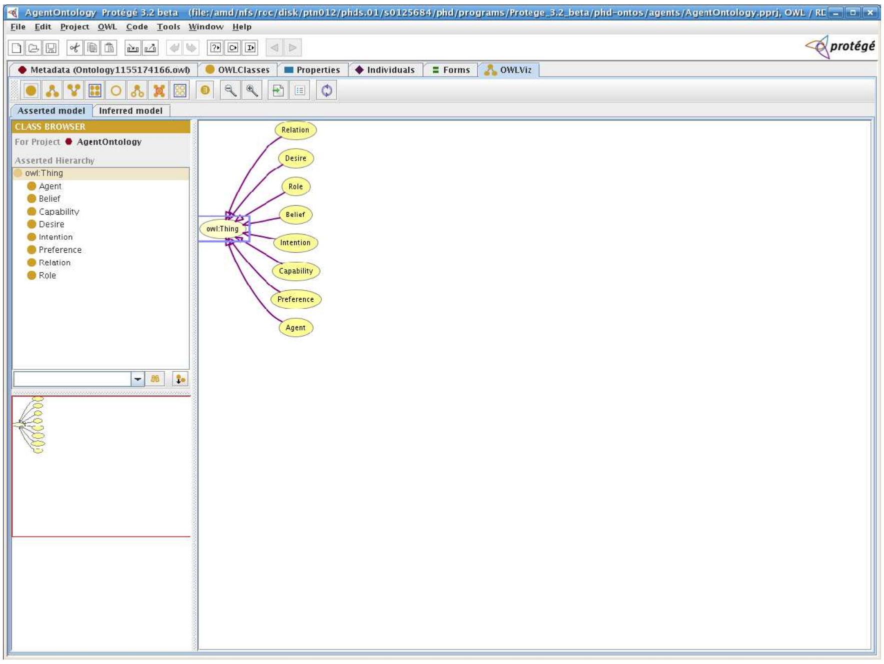


Figure A.4: Agent ontology.

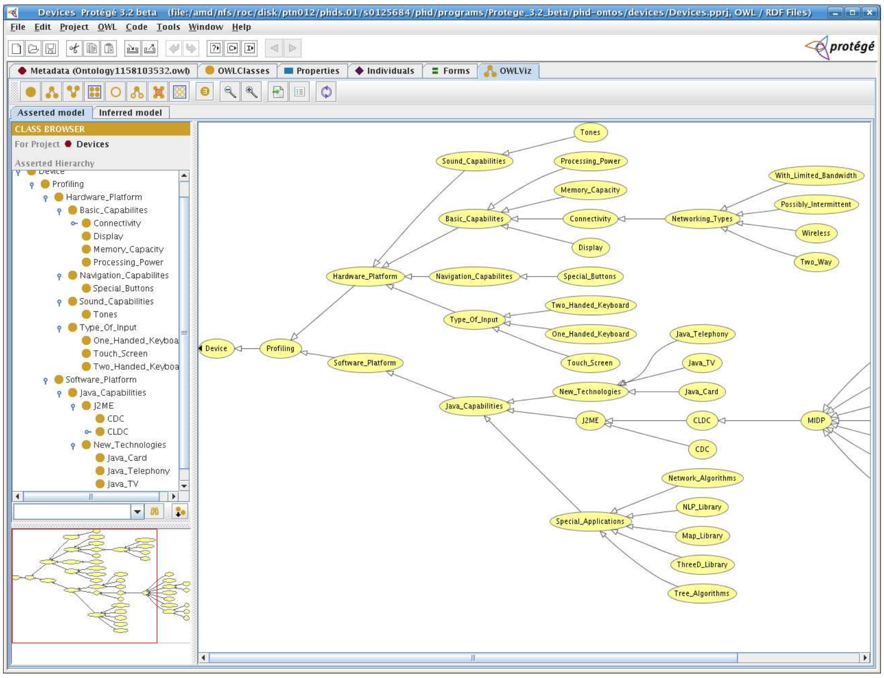


Figure A.5: Device ontology.

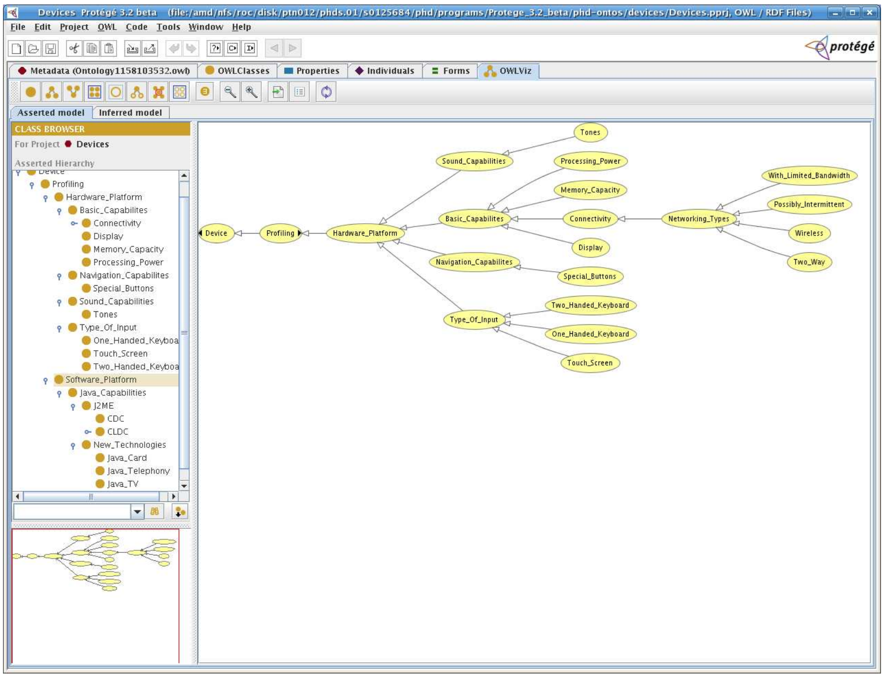


Figure A.6: Device ontology - Hardware platform focus.

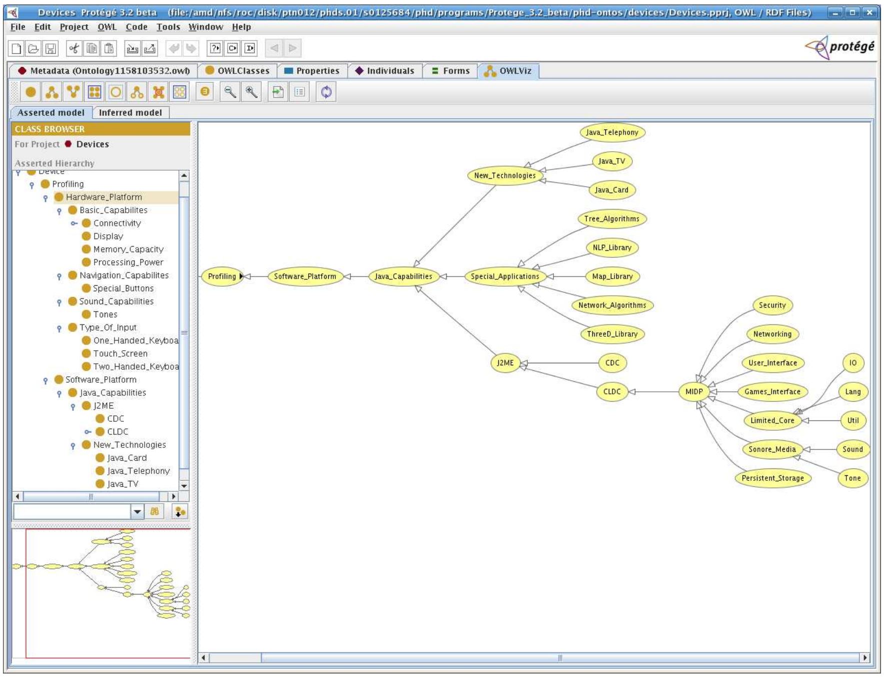


Figure A.7: Device ontology - Software platform focus.

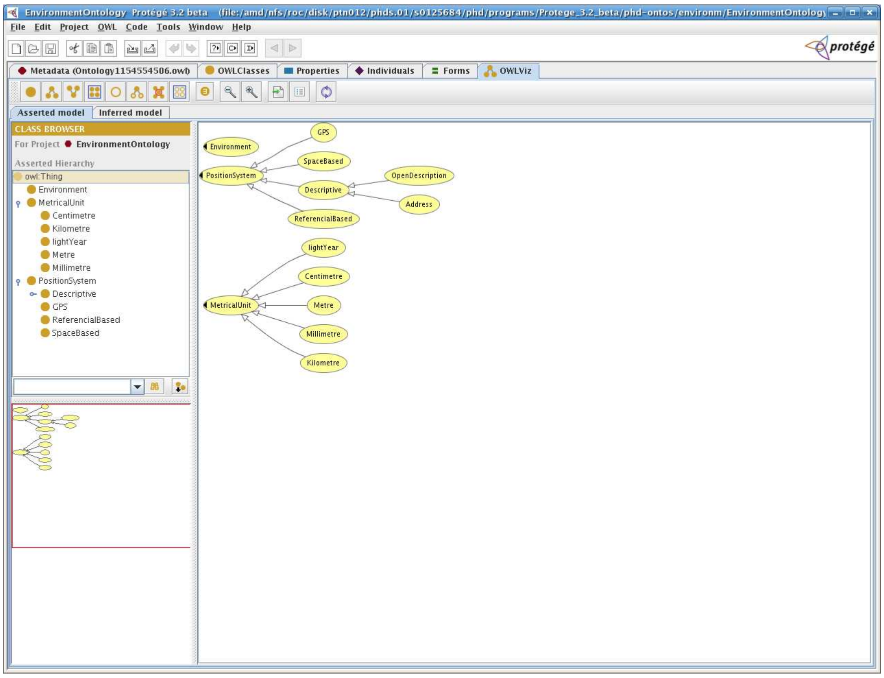


Figure A.8: Classes hierarchy of the environment ontology.

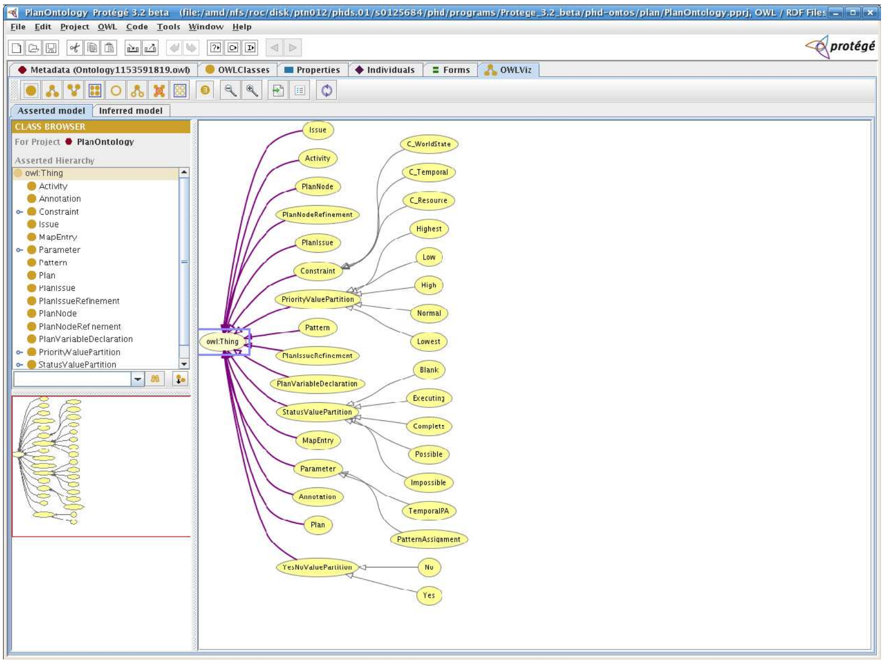


Figure A.9: Classes hierarchy of the planning ontology.

Appendix B

Rules′ Specification

The rules presented here describe the logic that has been used during the visualisation reasoning process. The implementation of such rules follows three ideas. First, because we are using the close world assumption, everything that is not in the knowledge base is false. Second, the rules basically manipulate facts, so that conclusion of rules implies insertions of new facts or changes in current facts. At last, we also have rules whose conclusion is an action to remove facts from the knowledge base. In other words, this action means that the fact, just removed, is no longer valid (close world assumption).

As an example, consider the rule conclusion "$\Rightarrow$ supports(d,m)", where d is a device and m is a modality. This rule conclusion means that the fact "the device d supports the modality m" must be inserted/updated in the knowledge base. In the same way, we can have "$\Rightarrow$ remove(supports(d,m))" to remove the fact from the base. In brief, all these rules must be understood as production system rules.

Examples of implementations of such rules can be seen in Appendix D and more details of this process in Chapter 8.

B.1 Device-restriction Rules

B.1.1 Basics

1. For every instance of the device class, this instance is able to support the textual and tabular visualisation modalities. In this way, we can say that both modalities are default options of visualisation for every device.

   $\forall$d,m Device(d) $\wedge$ ((m = Textual) $\vee$ (m = Tabular)) $\Rightarrow$ supports(d,m)

2. For every instance of the device class, if this instance has a GUI library, then it supports both graphical and temporal visualisation modalities. Note that temporal representations can be considered as a graphical representation whose one of the measure units is time.

   $\forall$d,m Device(d) $\wedge$ HasLibrary(d,GUI) $\wedge$ ((m = Graphic) $\vee$ (m = Temporal))
   $\Rightarrow$ supports(d,m)

B.1.2 Java Technology Semantic Based

1. For every instance of the device class, if this instance has Java capabilities and if this instance has a CLDC configuration, then it has available the MIDP profile.

   $\forall$d JavaEnabled(d) $\wedge$ HasConfiguration(d,CLDC) $\Rightarrow$ HasProfile(d,MIDP)

2. For every instance of the device class, if it has the MIDP profile, it obligatorily has the standard features of basic user interface, games interface, sonore media, networking and persistent storage.

   $\forall$d HasProfile(d,MIDP) $\Rightarrow$ HasLibrary(d,GUI) $\wedge$ HasLibrary(d,Games)
   $\wedge$ HasLibrary(d,Sonore) $\wedge$ HasLibrary(d,Networking) $\wedge$
   HasLibrary(d,Persistence-Storage)

3. For every instance of the device class, if it has the MIDP profile, then this instance supports the textual, tabular and graphical modality.

   $\forall$d HasProfile(d,MIDP) $\Rightarrow$ Supports(d,Textual) $\wedge$ Supports(d,Tabular) $\wedge$
   Supports(d,GUI)

4. For every instance of the device class, if it has the MIDP profile, and if the device has sound capabilities (hardware), then this instance supports the sonore modality.

   $\forall$d HasProfile(d,MIDP) $\wedge$ SoundEnabled(d) $\Rightarrow$ Supports(d,Sonore)

5. For every instance of the device class, if it has MIDP profile, and if the device has navigation capabilities (hardware), then this instance supports the use of pagination (independently of modality).

$\forall$d HasProfile(d,MIDP) $\wedge$ Supports(d,Navigation) $\Rightarrow$ Supports(d,Pagination)

6. For every instance of the device class, if it has the MIDP profile, then it supports special Java applications (for instance: Map, NLP, 3D, Network Algorithms or Tree Algorithms).

   $\forall$d HasProfile(d,MIDP) $\Rightarrow$ Supports(d,Especial-Java-Application)

7. For every instance of the device class, if it supports special Java applications, and it has a map library, then this instance supports the map based visualisation modality.

   $\forall$d Supports(d,Especial-Java-Application) $\wedge$ HasLibrary(MapLib)
   $\Rightarrow$ Supports(d,Map)

8. For every instance of the device class, if it supports special Java applications, and it has a NLP library, then this instance supports the natural language visualisation modality.

   $\forall$d Supports(d,Especial-Java-Application) $\wedge$ HasLibrary(NPLLib)
   $\Rightarrow$ Supports(d,NPL)

9. For every instance of the device class, if it supports special Java applications, and it has a 3D library, then this instance supports the three dimensional visualisation modalities (spatial representation and virtual reality).

   $\forall$d Supports(d,Especial-Java-Application) $\wedge$ HasLibrary(3DLib)
   $\Rightarrow$ Supports(d,Spatial) $\wedge$ Supports(d,Virtual-Reality)

10. For every instance of the device class, if it supports special Java applications, and it has a network algorithm library, then this instance supports the network visualisation modality.

    $\forall$d Supports(d,Especial-Java-Application) $\wedge$ HasLibrary(Net-Algorithm)
    $\Rightarrow$ Supports(d,Network)

11. For every instance of the device class, if it supports special Java applications, and it has a tree algorithm library, then this instance supports the tree visualisation modality.

$\forall$d Supports(d,Especial-Java-Application) $\wedge$ HasLibrary(Tree-Algorithms) $\Rightarrow$ Supports(d,Tree)

12. For every instance of the device class, if it has the MIDP profile, then it supports the new and advanced technologies supported by J2ME (for instance: Java Telephony, Java Card and Java TV).

$\forall$d,x NewTechnology(x) $\wedge$ HasProfile(d,MIDP) $\wedge$ Supports(J2ME,x) $\Rightarrow$ Supports(d,x)

B.1.3 Display x Sound x Navigation Semantic Based

1. For every instance of the device class, if this instance has support to generate sounds, then it supports the sonore visualisation modality.

$\forall$d,m Device(d) $\wedge$ SoundEnabled(d) $\wedge$ (m = Sonore) $\Rightarrow$ Supports(d,m)

2. For every instance of the device class, if this instance supports the graphic modality and the device display size is bigger than an specific constant value, then this instance supports the network and tree visualisation modality.

$\forall$d,m Supports(d,Graphic) $\wedge$ (DisplaySize(d) $>$ MinimalDisplaySize(m)) $\wedge$ ((m = Tree) $\vee$ (m=Network)) $\Rightarrow$ Supports(d,m)

3. The next rule refers to pagination (construction and change of more than one interface) and consequent navigation. For every instance of the device class, if this instance supports pagination, then it can support a multi-modality visualisation.

$\forall d,m_1,m_2$ Supports(d,Sonore) $\wedge$ Supports(d,m_1) $\wedge$ Supports(d,m_2) $\wedge$ $\neg(m_1 = m_2)$ $\Rightarrow$ MultiModality(d,m_1,m_2)

B.1.4 Advanced and New Technnologies Semantic Based

1. For every instance of the device class, if this instance has the OpenMap library, then it supports the map visualisation modality. Note that we are restricting the map libraries to just one option because is that library that we are using in our applications. However this restriction is not necessary at all.

   $\forall$d,m Device(d) $\wedge$ HasLibrary(d,OpenMap) $\wedge$ (m = Map) $\Rightarrow$ Supports(d,m)

2. For every instance of the device class, if this instance has OpenGL or DirectX among its libraries, so such an instance supports the spatial representations visualisation modality.

   $\forall$d,m Device(d) $\wedge$ (HasLibrary(d,OpenGL) $\vee$ HasLibrary(d,DirectX))
   $\wedge$ (m = Spatial) $\Rightarrow$ Supports(d,m)

3. For every instance of the device class, if this instance supports the sonore modality and it has the minimal processing power(PP) and memory requirements for a NLP application and it also has a NPL library installed, then such an instance supports the natural language modality.

   $\forall$d,m Supports(d,Sonore) $\wedge$ (ProcessingPower(d) $>$ MinimalPP(NLP))
   $\wedge$ (MemoryCapability(d) $>$ MinimalMemory(NLP)) $\wedge$ HasLibrary(d,NLP)
   $\wedge$ (m = Natural-Language) $\Rightarrow$ Supports(d,m)

4. For every instance of the device class, if this instance supports the spatial representation modality together with minimal requirements of processing power and memory capacity, then such a instance also supports the virtual reality (VR) visualisation modality.

   $\forall$d,m Supports(d,Spatial) $\wedge$ (ProcessingPower(d) $>$ MinimalPP(VR)) $\wedge$
   (MemoryCapability(d) $>$ MinimalMemory(VR)) $\wedge$ (m = VR)
   $\Rightarrow$ Supports(d,m)

5. For every instance of the device class, if this instance has not enough memory capability or processing power to support both VR and NPL modalities, then such a device can only support one of these visualisation modalities.

$\forall$d Sum(MinimalMemory(PLN),MinimalMemory(VR)) $>$ MemoryCapability(d) $\vee$ Sum(MinimalPP(NPL), MinimalPP(VR) $>$ ProcessingPower(d)) $\Rightarrow$ VRorNPL(d)

6. For every instance of the device class, if this instance does not support both VR and NPL modalities, if VR is currently supported then NPL should not be supported.

   $\forall$d Supports(d,VR) $\wedge$ VRorNPL(d) $\Rightarrow$ remove(Supports(d,NLP))

7. For every instance of the device class, if this instance does not support both VR and NPL modalities, if NLP is currently supported then VR should not be supported.

   $\forall$d Supports(d,NLP) $\wedge$ VRorNPL(d) $\Rightarrow$ remove(Supports(d,VR))

B.2 Planning Information-restriction Rules

B.2.1 Basics

1. For every instance of the plan class, if such an instance has a set of activities (sa), then every activity of this set is an activity of the plan.

   $\forall$p,a Plan(p) $\wedge$ ElementOf(sa,p) $\wedge$ Contains(sa,a) $\Rightarrow$ ActivityOf(a,p)

2. For every instance of the plan class, if such an instance has a set of issues (si), then every issue of this set is an issue of the plan.

   $\forall$p,i Plan(p) $\wedge$ ElementOf(si,p) $\wedge$ Contains(si,i) $\Rightarrow$ IssueOf(i,p)

3. For every instance of the plan class, if such an instance has a set of constraints (sc), then every constraint of this set is a constraint of the plan.

   $\forall$p,c Plan(p) $\wedge$ ElementOf(sc,p) $\wedge$ Contains(sc,c) $\Rightarrow$ ConstraintOf(c,p)

4. For every instance of the plan class, if such an instance has a set of annotations (sa), then every annotation of this set is an annotation of the plan.

   $\forall$p,a Plan(p) $\wedge$ ElementOf(sa,p) $\wedge$ Contains(sa,a) $\Rightarrow$ AnnotationOf(a,p)

B.2.2 Types of Planning Information

1. For every instance of the plan class, the information related to any of the plan element of this instance can be delivered via a textual, tabular, NLP or sonore representation.

   $\forall$p,e,m Plan(p) $\wedge$ ElementOf(e,p) $\wedge$ ((m = Textual) $\vee$ (m = Tabular) $\vee$ (m = NLP) $\vee$ (m = Sonore)) $\Rightarrow$ DisplayEnabled(e,m)

2. For every constraint of a plan, if the type of this constraint is temporal, then it can be visualised via a temporal representation.

   $\forall$p,c ConstraintOf(c,p) $\wedge$ type(c,Temporal) $\Rightarrow$ DisplayEnabled(p,c,Temporal)

3. For every constraint of a plan, if the type of this constraint is world-state, then it can be visualised via a map or virtual reality representations.

   $\forall$p,c ConstraintOf(c,p) $\wedge$ Type(c,World-State) $\Rightarrow$ DisplayEnabled(p,c,Map) $\wedge$ DisplayEnabled(p,c,VR)

4. For every constraint of a plan, if the type of this constraint is resource and if the resource has a geographic position, then it can be visualised via a map representation.

   $\forall$p,c ConstraintOf(c,p) $\wedge$ Type(c,Resource) $\wedge$ Has2dPosition(Object(c)) $\Rightarrow$ DisplayEnabled(p,c,Map)

5. For every activity of a plan, if this activity does not have a refinement, then it can be visualised via a network representation.

   $\forall$p,a ActivityOf(a,p) $\wedge$ $\neg$HasRefinement(a) $\Rightarrow$ DisplayEnabled(p,a,Network)

6. For every activity of a plan, if this activity has a refinement, then it can be visualised via a tree representation.

   $\forall$p,a ActivityOf(a,p) $\wedge$ hasRefinement(a) $\Rightarrow$ DisplayEnabled(p,a,Tree)

7. For every issue of a plan, if this issue does not have a refinement, then it can be visualised via a network representation.

   $\forall$p,i IssueOf(i,p) $\wedge$ $\neg$HasRefinement(i) $\Rightarrow$ DisplayEnabled(p,i,Network)

8. For every issue of a plan, if this issue has a refinement, then it can be visualised via a tree representation.

   $\forall$p,i IssueOf(i,p) $\wedge$ HasRefinement(i) $\Rightarrow$ DisplayEnabled(p,i,Tree)

B.2.3 Multi Modal Possibility

1. For every modality "m" enabled to display an element "e" of a plan "p", if the current device does not support such a modality, then this permission of visualisation is no longer valid. In other words, this rule eliminates the required ways of displaying an infomation that are not supported for the current device.

   $\forall$m,e,p displayEnabled(p,e,m) $\wedge$ $\neg$supports(d,m) $\Rightarrow$ remove(displayEnabled(p,e,m))

2. For every modality enabled to display an element "e" of a plan "p", if there are two of these modalities (m_1 and m_2) to display the same information, then this information can be visualised in a multi-mode way.

   $\forall$p,e,m_1,m_2 displayEnabled(p,e,m_1) $\wedge$ displayEnabled(p,e,m_2) $\wedge$ $\neg$(m_1 = m_2)
   $\Rightarrow$ multiVisualisation(p,e,m_1,m_2)

B.3 Agent-restriction Rules

B.3.1 Agents′ Preferences

1. If there is the possibility for a multi-visualisation of a plan information, and the user "u" has preference for one of the possible visualisation modalities, then the other option(s) are removed.

   $\forall$u,p,e,m_1,m_2 MultiVisualisation(p,e,m_1,m_2) $\wedge$ UserPreference(u,e,m_1)
   $\Rightarrow$ remove(DisplayEnabled(p,e,m_2))

B.4 Environment-restriction Rules

B.4.1 Location Based Awareness

1. For all instance of the plan class, if this plan has two constraints that refer to the same object and such constraints has latitude and longitude as attributes, then the object of these constraints has a 2D position.

   $\forall p,c_1,c_2\ \mathrm{ConstraintOf}(c_1,p) \wedge \mathrm{ConstraintOf}(c_2,p) \wedge (\mathrm{ObjectOf}(c_1) = \mathrm{ObjectOf}(c_2)) \wedge \mathrm{Attribute}(c_1,\mathrm{Latitude}) \wedge \mathrm{Attribute}(c_2,\mathrm{Longitude}) \Rightarrow \mathrm{Has2dPosition}(\mathrm{Object}(c_1))$

2. For all instance of the plan class, if this plan has a constraint object that has a 2D representation and such object is represented in other constraint whose attribute is altitude, then the object of this constraint has a 3D position

   $\forall p,c_1,c_2\ \mathrm{ConstraintOf}(c_1,p) \wedge \mathrm{ConstraintOf}(c_2,p) \wedge \mathrm{Has2dPosition}(\mathrm{Object}(c_1)) \wedge (\mathrm{ObjectOf}(c_1) = \mathrm{ObjectOf}(c_2)) \wedge \mathrm{Attribute}(c_2,\mathrm{Altitude})) \Rightarrow \mathrm{Has3dPosition}(\mathrm{Object}(c_1))$

Appendix C

Filtering Rules

One first observation for these rules is that they need to follow a sequence to be applied. This sequence accounts for giving the order of preference to the visualisation classes. In our application, this sequence is: *Special Structure*, *Complex Structure* and *Simple Structure*. A second observation is that, for our application, the last rule is never applied because the Tabular modality is one of the default modalities and it is relative to the *Complex Structure*. However, the last rule is important to cases where this assumption (Tabular as a default modality) is not taken.

1. The first filtering rule infers all the possible relations of the visualisation model via the concept of extension. So considering three model classes c_1, c_2 and c_3, if c_1 is a relative of c_2 and c_2 is relative of c_3, then c_1 is also relative of c_3.

   $\forall c_1,c_2,c_3\ \mathrm{Relative}(c_1,c_2) \wedge \mathrm{Relative}(c_2,c_3) \Rightarrow \mathrm{Relative}(c_1,c_3)$

2. The second rule gives preference to one modality that is relative to the *Special Structure* class. So, considering two visualisation instances, if one of these instances, for each plan element, is relative of the *Special Structure* class, then the other is removed from the base.

   $\forall v_1,v_2\ \mathrm{Visualisation}(v_1) \wedge \mathrm{Visualisation}(v_2) \wedge \mathrm{ElementType}(v_1,e) \wedge$
   $\mathrm{ElementType}(v_2,e) \wedge \mathrm{ModalityType}(v_1,m_1) \wedge \mathrm{ModalityType}(v_1,m_2)$
   $\wedge \neg(m_1{=}m_2) \wedge \mathrm{Relative}(m_1,\mathrm{SpecialStrucutre}) \Rightarrow \mathrm{remove}(\mathrm{Visualisation}(v_2))$

3. The third rule gives preference to one modality that is relative to the *Complex Structure* class. In fact the conditions of this rule will hold only if the conditions

of the previous rule does not hold. So, considering two visualisation instances, if one of these instances, for each plan element, is relative of the *Complex Structure* class, then the other is removed from the base.

$\forall v_1,v_2$ Visualisation(v_1) $\wedge$ Visualisation(v_2) $\wedge$ ElementType(v_1,e) $\wedge$ ElementType(v_2,e) $\wedge$ ModalityType(v_1,m_1) $\wedge$ ModalityType(v_1,m_2) $\wedge$ $\neg$(m_1=m_2) $\wedge$ Relative(m_1,ComplexStrucutre) $\Rightarrow$ remove(Visualisation(v_2))

4. The last rule is used only to ensure that if there are more that one visualisation whose modality is relative to the *Simple Structure* class, just one of these visualisations must hold. So, considering two visualisation instances, if one of these instances, for each plan element, is relative of the *Simple Structure* class, then the other (that will be also Simple structure when this rule is applied) is removed from the base.

   $\forall v_1,v_2$ Visualisation(v_1) $\wedge$ Visualisation(v_2) $\wedge$ ElementType(v_1,e) $\wedge$ ElementType(v_2,e) $\wedge$ ModalityType(v_1,m_1) $\wedge$ ModalityType(v_1,m_2) $\wedge$ $\neg$(m_1=m_2) $\wedge$ Relative(m_1,SimpleStrucutre) $\Rightarrow$ remove(Visualisation(v_2))

Appendix D

Rules in Object-Oriented JEOPS Syntax

```
public  ruleBase MultimodalityBase {
rule device01 { // Implements part of Rule B.1.1-1
      declarations
         Device d;
         Modality m;
      conditions
         m.isType("Textual");
         !m.isEnabled();
      actions
         m.setEnabled(true);
         modified(m);
}
rule device02 { // Implements part of Rule B.1.1-1
      declarations
         Device d;
         Modality m;
      conditions
         m.isType("Tabular");
         !m.isEnabled();
      actions
         m.setEnabled(true);
         modified(m);
```

```
}
rule device03 { // Implements part of Rule B.1.1-2
    declarations
        Device d;
        Modality m;
    conditions
        d.hasLibrary("GUI");
        m.isType("Graphic");
        !m.isEnabled();
    actions
        m.setEnabled(true);
        modified(m);
}
rule device04 { // Implements part of Rule B.1.1-2
    declarations
        Device d;
        Modality m1;
        Modality m2;
    conditions
        m1.isType("Graphic");
        m1.isEnabled();
        m2.isType("Temporal");
        !m2.isEnabled();
    actions
        m2.setEnabled(true);
        modified(m2);
}
rule device05 { // Implements Rule B.1.3-1
    declarations
        Device d;
        Modality m;
    conditions
        d.hasResource("Sound");
        m.isType("Sonore");
        !m.isEnabled();
```

```
    actions
        m.setEnabled(true);
        modified(m);
}
rule device06 { // Implements part of Rule B.1.3-2
    declarations
        Device d;
        Modality m1;
        Modality m2;
    conditions
        m1.isType("Graphic");
        m1.isEnabled();
        m2.isType("Network");
        !m2.isEnabled();
        d.biggerThan(m2.getMinimalScreenSize());
    actions
        m2.setEnabled(true);
        modified(m2);
}
rule device07 { // Implements part of Rule B.1.3-2
    declarations
        Device d;
        Modality m1;
        Modality m2;
    conditions
        m1.isType("Graphic");
        m1.isEnabled();
        m2.isType("Tree");
        !m2.isEnabled();
    d.biggerThan(m2.getMinimalScreenSize());
    actions
        m2.setEnabled(true);
        modified(m2);
}
rule device08 { // Implements Rule B.1.4-1
```

```
    declarations
        Device d;
        Modality m;
    conditions
        m.isType("Map");
        !m.isEnabled();
        d.hasLibrary("Map");
    actions
        m.setEnabled(true);
        modified(m);
}
rule device09 { // Implements part of Rule B.1.4-2
    declarations
        Device d;
        Modality m;
    conditions
        m.isType("Spatial");
        !m.isEnabled();
        d.hasLibrary("OpenGL");
    actions
        m.setEnabled(true);
        modified(m);
}
rule device10 { // Implements part of Rule B.1.4-2
    declarations
        Device d;
        Modality m;
    conditions
        m.isType("Spatial");
        !m.isEnabled();
        d.hasLibrary("DirectX");
    actions
        m.setEnabled(true);
        modified(m);
}
```

```
rule device11 { // Implements Rule B.1.4-4
    declarations
        Device d;
        Modality m1;
        Modality m2;
    conditions
        m1.isType("Spatial");
        m1.isEnabled();
        m2.isType("VirtualReality");
        !m2.isEnabled();
        d.getProcessingPower() > m2.getMinimalProcessingPower();
        d.getMemory() > m2.getMinimalMemory();
    actions
        m2.setEnabled(true);
        modified(m2);
}
rule device12 { // Implements Rule B.1.4-5
    declarations
        Device d;
        Modality m1;
        Modality m2;
    conditions
        m1.isType("Sonore");
        m1.isEnabled();
        m2.isType("NaturalLanguage");
        !m2.isEnabled();
        d.hasLibrary("NPL");
        d.getProcessingPower() > m2.getMinimalProcessingPower();
        d.getMemory() > m2.getMinimalMemory();
    actions
        m2.setEnabled(true);
        modified(m2);
}
rule plan01 { // Implements Rule B.2.1-1
    declarations
```

```
        VPlan p;
    conditions
        p.hasActivities();
    actions
        PlanElement pe = new PlanElement(p,“Activities”,p.getActivities());
        assertt(pe);
        p.removeActivities();
        modified(p);
}
rule plan02 { // Implements Rule B.2.1-2
    declarations
        VPlan p;
    conditions
        p.hasIssues();
    actions
        PlanElement pe = new PlanElement(p,“Issues”,p.getIssues());
        assertt(pe);
        p.removeIssues();
        modified(p);
}
rule plan03 { // Implements Rule B.2.1-3
    declarations
        VPlan p;
    conditions
        p.hasConstraints();
    actions
        PlanElement pe = new PlanElement(p,“Constraints”,p.getConstraints());
        assertt(pe);
        p.removeConstraints();
        modified(p);
}
rule plan04 { // Implements Rule B.2.1-4
    declarations
        VPlan p;
    conditions
```

```
    p.hasAnnotations();
  actions
    PlanElement pe = new PlanElement(p,“Annotations”,p.getAnnotations());
    assertt(pe);
    p.removeAnnotations();
    modified(p);
}
rule plan05 { // Implements Rule B.2.2-1
  declarations
    PlanElement pe;
    VPlan p;
  conditions
    pe.isFrom(p);
  actions
    Visualisation v1 = new Visualisation(p,pe,“Textual”);
    Visualisation v2 = new Visualisation(p,pe,“Tabular”);
    Visualisation v3 = new Visualisation(p,pe,“NaturalLanguage”);
    Visualisation v4 = new Visualisation(p,pe,“Sonore”);
    assertt(v1);
    assertt(v2);
    assertt(v3);
    assertt(v4);
}
rule plan06 { // Implements Rule B.2.2-2
  declarations
    PlanElement pe;
    VPlan p;
    Visualisation v;
  conditions
    pe.areElements(“Constraints”);
    pe.hasSubType(“temporal”);
  actions
    Visualisation v = new Visualisation(p,pe,“Temporal”);
    assertt(v);
    retract(pe);
```

```
}
rule plan07 { // Implements Rule B.2.2-3
    declarations
        PlanElement pe;
        VPlan p;
    conditions
        pe.areElements(“Constraints”);
        pe.hasSubType(“world-state”);
    actions
        Visualisation v1 = new Visualisation(p,pe,“Map”);
        Visualisation v2 = new Visualisation(p,pe,“VirtualReality”);
        assertt(v1);
        assertt(v2);
        retract(pe);
}
rule plan08 { // Implements Rule B.2.2-5
    declarations
        PlanElement pe;
        VPlan p;
    conditions
        pe.areElements(“Activities”);
        !pe.hasRefinement();
    actions
        Visualisation v = new Visualisation(p,pe,“Network”);
        assertt(v);
}
rule plan09 { // Implements Rule B.2.2-6
    declarations
        PlanElement pe;
        VPlan p;
    conditions
        pe.areElements(“Activities”);
        pe.hasRefinement();
    actions
        Visualisation v = new Visualisation(p,pe,“Tree”);
```

```
        assertt(v);
}
rule plan10 { // Implements Rule B.2.2-7
    declarations
        PlanElement pe;
        VPlan p;
    conditions
        pe.areElements("Issues");
        !pe.hasRefinement();
    actions
        Visualisation v = new Visualisation(p,pe,"Network");
        assertt(v);
}
rule plan11 { // Implements Rule B.2.2-8
    declarations
        PlanElement pe;
        VPlan p;
    conditions
        pe.areElements("Issues");
        pe.hasRefinement();
    actions
        Visualisation v = new Visualisation(p,pe,"Tree");
        assertt(v);
}
rule Other01 { // Implements Rule B.2.3-1
    declarations
        Modality m;
        Visualisation v;
    conditions
        m.isType(v.getModalityType());
        !m.isEnabled();
    actions
        retract(v);
}
rule Other02 { // Implements Rule B.3.1-1
```

```
    declarations
        Visualisation v1;
        Visualisation v2;
        Agent a;
    conditions
        (v1.getElementType()).equals(v2.getElementType());
        !(v1.getModalityType()).equals(v2.getModalityType());
        a.prefers(v1);
    actions
        retract(v2);
}
}
```

Appendix E

Publications

E.1 By Chronological Order

- Lino, N., Tate, A., Siebra, C. and Chen-Burger, Y. (2003) Delivering Intelligent Planning Information to Mobile Devices Users in Collaborative Environments, Workshop on Artificial Intelligence, Information Access and Mobile Computing (AI-IA-MC) at the International Joint Conference on Artificial Intelligence (IJCAI-03), Acapulco, Mexico, August 2003.

- Lino, N. and Tate, A. (2004) M-Planning: A Mobile Tool to Support Collaborative Planning. Proceedings of the International Conference on Artificial Intelligence and Applications (AIA-2004), as part of the Twenty-Second IASTED International Multi-Conference on Applied Informatics, Innsbruck, Austria, February 2004.

- Lino, N. (2004) An Integrated Ontology Set and Reasoning Mechanism for Multi-Modality Visualisation Destined to Collaborative Planning Environments. Student Paper for Doctoral Consortium at the Fourteenth International Conference on Automated Planning and Scheduling (ICAPS-2004), Whistler, British Columbia, Canada. 3-7 June 2004.

- Siebra, C., Tate, A. and Lino, N. (2004) Planning and Representation of Joint Human-Agent Space Missions via Constraint-Based Models, Fourth International Workshop on Planning and Scheduling for Space (IWPSS-04), Darmstadt, Germany, 23-25 June 2004.

- Lino, N. and Tate, A. (2004) A Visualisation Approach for Collaborative Planning Systems Based on Ontologies, in Proceedings of the 8th International Conference on Information Visualisation (IV 2004), London, UK, 14-16 July 2004, IEEE Computer Society Press, 2004.

- Lino, N., Tate, A., Siebra, C. and Chen-Burger, Y. (2004) Improving Semantics in Mobile Devices Profiling: A Model Conceptual Formalisation and Ontology Specification, Workshop on Semantic Web Technology for Mobile and Ubiquitous Applications at the 3rd International Semantic Web Conference, Hiroshima, Japan, 7-11 November 2004.

- Lino, N., Tate, A., and Chen-Burger, Y-H. (2005) Semantic Support for Visualisation in Collaborative AI Planning, in Proceedings of the Workshop on The Role of Ontologies in Planning and Scheduling, at the International Conference on Automated Planning and Scheduling (ICAPS), June 2005, Monterey, California, USA.

Bibliography

[Aamodt and Plaza, 1994] Aamodt, A. and Plaza, E. (1994). Case-Based Reasoning: Foundational Issues, Methodological Variations and System Approaches. *Artificial Intelligence Communications*, 7(1):39-52.

[Ai-Chan et al., 2004] Ai-Chang, M., Bresina, J., Charest, L., Chase, A., Hsu, J.C.-J., Jonsson, A., Kanefsky, B., Morris, P., Kanna Rajan, Yglesias, J., Chafin, B.G., Dias, W.C., Maldague, P.F. (2004). MAPGEN: Mixed-Initiative Planning and Scheduling for the Mars Exploration Rover Mission. *Intelligent Systems*, Volume 19, Issue 1, Jan-Feb 2004, pp. 8-12.

[Allen et al., 2001a] Allen, J., Byron, D., Dzikovska, M., Ferguson, G., Galescu, L., and Stent, A. (2001). Towards conversational human-computer interaction. *In AI Magazine, 22(4), Winter 2001*, pp. 27-37.

[Allen et al, 2001b] Allen, J., Ferguson, G. and Stent, A. (2001). An architecture for more realistic conversational systems. *In Proceedings of Intelligent User Interfaces 2001 (IUI-01)*, Santa Fe, NM, USA.

[Alonso at al., 1998] Alonso, D., Rose, A., Plaisant, C., and Norman, K. (1998). Viewing Personal History Records: A Comparison of Tabular Format and Graphical Presentation Using LifeLines. *In Behavior and Information Technology*, pp. 249-262.

[Antoniou and Harmelen, 2004] Antoniou, G., Harmelen, F. (2004). A Semantic Web Primer. *The MIT Press, Cambridge*, Massachusetts, April 2004, ISBN 0-262-01210-3.

[Baus et al., 2002] Baus, J., Kruger, A., Wahlster, W. (2002). A Resource-Adaptive Mobile Navigation System. *In Proceedings of ACM Intelligent User Interfaces - IUI 2002.*

[Bacchus and Ady, 2001] Bacchus, F. and Ady, M. (2001). Planning with Resources and Concurrency: A Forward Chaining Approach. In Proceedings of the International Joint Conferences on Artificial Intelligence - IJCAI 2001

[Blythe et al., 2003] Blythe, J., Deelman, E., Gil, Y. and Kesselman, C. (2003). Transparent Grid Computing: A Knowledge-Based Approach *Conference on Innovative Applications of Artificial Intelligence (IAAI) 2003*

[Berners-Lee and Miller, 2002] Berners-Lee, T. and Miller, E. (2002). The Semantic Web lifts off. *ERCIM News No. 51*, October 2002.

[Bertin, 1967/1983] Bertin, J. (1967). Semiology of Graphics:Diagrams, Networks, Maps. *W. J. Berg, Trans.*, Madison, WI : University of Wisconsin Press.

[Blum and Furst, 1997] Blum, A. and Furst, M. (1997). Fast Planning Through Planning Graph Analysis. *Artificial Intelligence*, pp. 90:281-300.

[Brachman and Levesque, 2004] Brachman, R. J., and Levesque, H. J. (2004). Knowledge Representation and Reasoning. *Morgan Kaufmann*, San Francisco, CA, 2004.

[Brafman and Hoffmann, 2004] Brafman, R. and Hoffmann, J. (2004). Conformant Planning via Heuristic Forward Search: A New Approach *In Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling*, Whistler, British Columbia, Canada.

[Burke and Petrovic, 2002] Burke, E. K. and Petrovic, S. (2002). Recent Research Directions in Automated Timetabling. *In European Journal of Operational Research - EJOR*, Vol. 140, No. 2, pp. 266-280.

[Burkhard 2004] Burkhard, R. A. (2004). Learning from Architects: The Difference between Knowledge Visualisation and Information Visualisation. *In Proceedings of the Eight International Conference on Information Visualisation (IV-04)*, London, England, UK.

[Card et al., 1991] Card, S. K., Robertson, G. G., and Mackinlay, J. D. (1991). The Information Visualizer: An Information Workspace. *In Proceedings of the ACM Conference on Human Factors in Computing Systmes - CHI 1991*, London, England, UK.

[Card et al., 1999] Card, S. K., Mackinlay, J. D., and Shneiderman, B. (1999). Readings in Information Visualization, Using Vision to Think. Los Altos, CA, Morgan Kaufmann.

[Caruana and Niculescu, 2006] Caruana, R. and Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. *Proceedings of the 23rd international conference on Machine learning*, pp. 161-168, Pittsburgh, Pennsylvania, USA.

[Casner, 1991] Casner, S. (1991). Task-Analytic Approach to the Automated Design of Graphic Presentation *ACM Transactions on Graphics*, 10(2, April), pp. 111-151.

[Champandard, 2003] Champandard, A. (2003). AI Game Development: Synthetic Creatures with Learning and Reactive Behaviors. *New Riders Publisher*, 2003.

[Cheverst et al., 2000] Cheverst, K., Davies, N., Mitchell, K., Friday A., Efstratiou C. (2000). Developing a Context-aware Eletronic Tourist Guide: Some Issues and Experiences. *In Proceedings of the ACM Conference on Human Factors in Computing Systmes - CHI 2000*.

[Chien et al., 1999] Chien, S., Knight, R., Stechert, A., Sherwood, R. and Rabideau, G. (1999). Using Iterative Repair to Increase the Responsiveness of Planning and

Scheduling for Autonomous Spacecraft. *In Proceedings of the International Joint Conferences on Artificial Intelligence - IJCAI 1999 Workshop on Scheduling and Planning meet Real-time Monitoring in a Dynamic and Uncertain World*, Stockholm, Sweden.

[Chien et al., 2000] Chien, S., Rabideau, G., Knight, R., Sherwood, R., Engelhardt, B., Mutz, D., Estlin, T., Smith, B., Fisher, F., Barrett, T., Stebbins, G., Tran, D. (2000). ASPEN - Automating Space Mission Operations using Automated Planning and Scheduling. *SpaceOps 2000*, Toulouse, France, FR.

[Cholewiak and Beede, 2005] Cholewiak, R. W. and Beede, K. (2005). The Representation of Space through Static and Dynamic Tactile Displays. *Virtual Reality International Conference in conjunction with the Human Computer Interaction 2005*, Las Vegas, NE.

[Chuah et al., 1995] Chuah, M., Roth, S., Mattis, J. and Kolojejchick, J. (1995). SDM: Selective Dynamic Manipulation of Visualisation. *In Proceedings of the ACM Symposium on User Interface Software and Technology - UIST 1995*, pp. 61-70.

[Currie and Tate, 1991] Currie, K. and Tate, A. (1991). O-Plan: the Open Planning Architecture. Artificial Intelligence Vol. 52, pp 49-86.

[Davis et al., 1977] Davis, R., Buchanan, B. G., and Shortliffe, E. H. (1977). Production rules as a representation for a knowledge-based consultation program. *Artificial Intelligence*, 8:15–45, 1977.

[Derthick and Roth, 2001] Derthick, M. and Roth, S. (2001). Example-based generation of custom data analysis appliances. *In Proceedings of the ACM Intelligent User Interfaces IUI 2001.*

[Do and Kambhampati, 2003] Do, M. B. and Kambhampati, S. (2003). Sapa: A Scalable Multi-Objective Metric Temporal Planner. *Journal of Artificial Intelligence Research*. To appear.

[Doherty and Kvarnstrm, 2001] Doherty, P. and Kvarnstrm, J. (2001). TALplanner: A Temporal Logic Based Planner. *AI Magazine*, Fall Issue.

[Edelkamp and Helmert, 2001] Edelkamp, S., Helmert, M. (2001). The Model Checking Integrated Planning System (MIPS). *In AI-Magazine*.

[Elting et al., 2002] Elting, C., Zwickel, J., Malaka, R. (2002). Device-Dependant Modality Selection for User-Interfaces An Empirical Study. *In Proceedings of ACM Intelligent User Interfaces - IUI 2002*.

[Encarnacao and Kirste, 2000] Encarnacao, J. L., Kirste, T. (2000). Beyond the Desktop: Natural Interaction and Intelligent Assistance for the Everyday Life. *In Altag der Zukunft Informationstechnik Verandert Unser Leben*.

[Erol et al., 1994] Erol, K., Nau, D. and Hendler, J. (1994). HTN Planning: Complexity and Expressivity. *Proceedings of the Twelfth National Conference on Artificial Intelligence*, Seattle, Washington, USA, pp.1123-1128.

[Feiner and Beshers, 1990] Feiner, S. K., and Beshers, C. (1990). Visualizing n-Dimensional Virtual Worlds with n-Vision. *Computer Graphics, 24(2)*, pp.37-38.

[Ferguson, 2000] Ferguson, G. (2000). TRIPS Pacifica Movie. `http://www.cs.rochester.edu/research/cisd/projects/trips/movies/TRIPS-98v4.0/`.

[Figueira and Ramalho, 2000] The Java Embedded Object Production System. *Lecture Notes in Artificial Intelligence*, v. 1952, pp. 52-61, Springer Verlag.

[Fikes and Nilsson, 1971] Fikes, R. E. and Nilsson, N. J. (1971). STRIPS: a new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3-4):189-208.

[Fluit at al., 2002] Fluit, C., Sabou, M. and Harmelen, F. (2002). Ontology-based Information Visualisation. *In Proceedings of Visualising the Semantic Web - VSW 2002*, Springer Verlag, Vladimir Geroimenko Editor.

[Forgy, 1982] Forgy, C. (1982). Rete: A Fast Algorithm for the Many Pattern"/" Many Object Pattern Match Problem. *Artificial Inteligence*, 19:17-37.

[Fox and Long, 2001] Fox, M. and Long, D. (2001). An Extension to PDDL for Expressing Temporal Planning Domains. *Technical Report, Department of Computer Science, University of Durham*, Durham, UK.

[Geroimenko and Chen, 2006] Geroimenko, V., Chen, C. (Eds) (2006). Visualising the Semantic Web: XML-based Internet and Information Visualisation. *Springer-Verlag*, 2006.

[Ghallab et al., 2004] Ghallab, M., Nau, D. and Traverso, P. (2004) Automated Planning: Theory and Practice. *Morgan Kaufmann*, May 2004.

[Gil et al., 2004] Gil, Y., Deelman, E., Blythe, J., Kesselman, C. and Tangmunarunkit, H. (2004). Artificial Intelligence and Grids: Workflow Planning and Beyond. *IEEE Intelligent Systems, Special Issue on E-Science*.

[Gupta et al., 1986] Gupta, A., Forgy, C., Newell, A. and Wedig, R. (1986). Parallel algorithms and architectures for rule-based systems. *Proceedings of the 13th annual international symposium on Computer architecture*, pp. 28-37, Tokyo, Japan.

[Ham and Wijk, 2002] Ham, F. and Wijk, J.J. (2002). Beamtrees: Compact Visualization of Large Hierarchies. *Proceedings of the IEEE Conference Information Visualization 2002, IEEE CS Press*, pp. 93-100.

[Haarslev and Moller, 2003] Haarslev, V. and Moller, R. (2003). Racer: A Core Inference Engine for The Semantic Web. *Proceedings of the Second International Workshop on Evaluation of Ontology-based Tools*, Semantic Web Conference, Sanibel Island, Florida, USA, pp. 27-36.

[Hannebauer and Geske, 1999] Hannebauer, M. and Geske, U. (1999). Coordinating Distributed CLP-Solvers in Medical Appointment Scheduling. *In Proceedings of the Twelfth International Conference on Applications of Prolog (INAP-99)*, Tokyo, Japan, pp. 117-125.

[Hinckley et al., 2004] Hinckley, K., Ramos, G., Guimbretiere, F., Baudisch, P. and Smith, M. (2004). Stitching: Pen Gestures that Span Multiple Displays. *ACM Advanced Visual Interfaces (AVI 2004)*.

[Hoffmann, 2001] Hoffmann, J. (2001). FF: The Fast-Forward Planning System. *In AI Magazine*, Volume 22, Number 3, pp. 57-62.

[Horrocks 2002] Horrocks, I. (2002). DAML+OIL: A Description Logic for the Semantic Web. *IEEE Bulletin of the Technical Committee on Data Engineering.*

[Jonsson, 1999] Jonsson, A., et al. (1999). Next Generation Remote Agent Planner. *In Proceedings of 5th International Symposium Artificial Intelligence, Robotics and Automation in Space (iSAIRAS99)*, European Space Agency Publications Division, pp. 363-370.

[Kambhampati, 2002] Kambhampati, S. (2002). SAPA: A Forward Chaining Heuristic Metric Temporal Planner. `http://rakaposhi.eas.asu.edu/sapa.html/`.

[Karp et al., 1994] Karp, P. D., Lowrance, J. D., Strat, T. M. and Wilkins, D. E. (1994). The Grasper-CL Graph Management System. *LISP and Symbolic Computation: An International Journal*, Kluwer Academic Publishers, vol. 7, pp. 251-290.

[Kautz and Selman, 1998a] Kautz, H. and Selman, B. (1998). The Role of Domain-Specific Knowledge in the Planning as Satisfiability Framework. *In Proceedings of the 4th International Conference on Artificial Intelligence Planning Systems (AIPS 1998)*, Pittsburgh, PA, USA.

[Kautz and Selman, 1998b] Kautz, H. and Selman, B. (1998). BLACKBOX: A New Approach to the Application of Theorem Proving to Problem Solving. *Working notes of the Workshop on Planning as Combinatorial Search (AIPS-98)*, Pittsburgh, PA, USA.

[Kerpedjiev et al., 1998] Kerpedjiev, S., Carenini, G., Green, N., Moore, J., and Roth, S.F. (1998). Saying It In Graphics: from Intentions to Visualizations. *Proceedings of the IEEE Symposium on Information Visualization (InfoVis '98)*, pp. 97 - 101.

[Kitano and Tadokoro, 2001] Kitano, H. and Tadokoro, S. (2001). RoboCup Rescue: A Grand Challenge for Multiagent and Intelligent Systems. *AI Magazine*, 22(1):39-52.

[Knublauch et al., 2004] Knublauch, H., Fergerson, R. W., Noy, N. F., and Musen, M. A. (2004). The Protg OWL Plugin: An Open Development Environment for Semantic Web Applications *Third International Semantic Web Conference*, Hiroshima, Japan, 2004.

[Larkin and Simon, 1987] Larkin, J., and Simon, H. A. (1987). Why a Diagram is (Sometimes) Worth Ten Thousand Words. *Cognitive Science, 11(1)*, pp. 65-99.

[Li et al., 2005] Li, Y., Hinckley, K., Guan, Z., Landay, J. A. (2005). Experimental Analysis of Mode Switching Techniques in Pen-based User Interfaces. *CHI 2005*, pp. 461-470.

[Lino et al., 2003] Lino, N., Tate, A., Siebra, C. and Chen-Burger, Y. (2003). Delivering Intelligent Planning Information to Mobile Devices Users in Collaborative Environments. *Workshop on Artificial Intelligence, Information Access and Mobile Computing (AI-IA-MC) at the International Joint Conference on Artificial Intelligence (IJCAI-03)*, Acapulco, Mexico, pp.

[Lino et al., 2004] Lino, N., Tate, A., and Chen-Burger, Y. (2004) Improving Semantics in Mobile Devices Profiling: A Model Conceptual Formalisation and Ontology Specification *Workshop on Semantic Web Technology for Mobile and Ubiquitous Applications at the 3rd International Semantic Web Conference*, Hiroshima, Japan.

[LiSA Team, 2003] LiSA Team. (2003). LiSA - A Library of Scheduling Algorithms. `http://lisa.math.uni-magdeburg.de/index.php`.

[Lohse et al., 1994] Lohse, G., Biolsi, K., Walker, N., and Rueter, H. (1994). A Classification of Visual Representations. *In Communications of the ACM, 37 (12)*, pp. 36-49.

[Long and Fox, 2003] Long, D. and Fox, M. (2003). Exploiting a Graphplan Framework in Temporal Planning. *In Proceedings of the 13th International Conference on Automated Planning and Scheduling (ICAPS)*, Trento, Italy.

[Long et al., 1996] Long, S., Kooper, K., Abowd, G. D., Atkeson, C. G. (1996). Rapid Prototyping of Mobile Context-Aware Applications: The Cyberguide Case Study. *In Proceedings of the 2nd ACM International Conference on Mobile Computing and Networking*.

[Mackinlay, 1986] Mackinlay, J. D.. (1986). Automatic Design of Graphical Presentations. *Unpuplished Ph.D. Thesis, Stanford University*.

[Macrofocus, 2005] Macrofocus. (2005). SurveyVisualizer. `Url: http://www.macrofocus.com/solutions/market_research.html`.

[Malaka and Zipf, 2000] Malaka, R., Zipf, A. (2000). Deep Map Challenging IT Research in the Framework of a Tourist Information System. *In Proceedings Information and Communication Technologies in Tourism*, Springer, Wien.

[Maldague et al., 1997] Maldague, P., et al. (1997). APGEN: A Multi-Mission Semi-Automated Planning Tool. *In Proceeding of the 1st International Workshop Planning and Scheduling for Space*, AAAI Press, 1997.

[Matkovic et al., 2002] Matkovic, K., Hauser, H., Sainitzer, R., Grller, M. E. (2002). Process Visualization using Levels of Detail. *Published in the Proceedings of the IEEE Symposium on Information Visualization 2002 (InfoVis 2002)*, Oct. 28-29, 2002, Boston, MA, pp. 67-70.

[Myers, 1999] Myers, Brad A. (1999). User Interface Management Systems. *Wiley Encyclopedia of Electrical and Electronics Engineering*, Volume 23. John G. Webster, editor. New York: John Wiley and Sons., pp. 42-58.

[Myers et al., 2002] Myers, K. L., Tyson, W. M., Wolverton, M. J., Jarvis, P. A., Lee, T. J., and desJardins, M. (2002). PASSAT: A User-centric Planning Framework. *In Proceedings of the the 3rd International NASA Workshop on Planning and Scheduling for Space*, Houston, TX, October, 2002.

[McCluskey and Simpson, 2004] McCluskey, T. L. and Simpson, R. (2004). Knowledge Formulation for AI Planning. *In Proceedings of 4th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2004)*, Whittlebury Hall, Northampton-shire, UK, pp. 459-465.

[McCormick and DeFanti, 1987] McCormick, B. H. and DeFanti, T.A. (1987). Visualization is Scientific Computing *Computer Graphics, 21 (6 November).*

[McDermott et al., 1998] McDermott, D., Ghallab, M., Howe, A., Knoblock, C., Ram, A., Veloso M., Weld, D. and Wilkins, D. (1998). PDDL - The Planning Domain Definition Language. *Technical Report CVC TR-98-003/DCS TR-1165*, Yale Center for Computational Vision and Control.

[Moran et al., 1997] Moran, D., Cheyer, A., Julia, L., and Park, S. (1997). Multimodal user interfaces in the Open Agent Architecture. *Proceedings of the International Conference on Intelligent User Interfaces (IUI97)*, Orlando, Florida, pp. 61-68.

[Muldowney and Landrum, 2000] Muldowney, T. and Landrum, E. (2000). The Jabber Programmers Guide: A Comprehensive Snap-shot of Jabber. *Jabber Software Foundation, Inc.*, Boston, MA, USA.

[Nguyen and Huang, 2002] Nguyen, Q.V.; Huang, M. L. (2002). A Space-Optimized Tree Visualization. *In the Proceedings of the IEEE Symposium on Information Visualization (InfoVis2002)*, pp. 85-92, Boston, Massachussets, USA.

[Nixon et al., 2000] Nixon, A., Levine, J., and Tate, A. (2000). A Limited Media Interface to an AI Planning System. *In Proceedings of the Nineteenth Workshop of the UK Planning and Scheduling Special Interest Group (PLANSIG 2000)*, Open University, Milton Keynes, UK.

[Not et al., 1998] Not, E., Petrelli, D., Sarini, M., Stock, O., Strapparava, C., Zancanaro, M. (1998). Hypernavigation in the Physical Space: Adapting Presentation to the User and to the Situational Context. *The New Review of Hypermidia and Multimedia.*

[Oxygen Project , 2003] Oxygen Project. (2003) Oxygen Project at MIT. `http://oxygen.lcs.mit.edu/Overview.html`.

[Parrod and Valera, 1993] Parrod, Y. and Valera, S. (1993). Optimum-AIV, A Planning Tool for Spacecraft AIV. *In Preparing for the Future*, Vol. 3, No. 3, European Space Agency, pp. 7-9.

[Polyak and Tate, 1997] Polyak, S. and Tate, A. (1997). Analysis of Candidate PSL Process/Plan Representations. *AIAI-PR-66, Artificial Intelligence Applications Institute (AIAI)*, Edinburgh, 1997.

[Rao and Georgeff, 1995] Rao, A. S., Georgeff, M. P. (1995). BDI Agents: From Theory to Practice. *Proceedings of the First International Conference on Multiagent Systems*, San Francisco, EUA.

[Reece et al., 1993] Reece,G. A.; Tate, A.; Brown, D. I.; Hoffman, M. and Burnar, R. E. (1993). The PRECiS Environment *Proceedings of the National Conference on Artificial Intelligence (AAAI-93), ARPA-RL Planning Initiative Workshop*, Washington, DC, 1993.

[Robertson et al., 1989] Robertson, G. G., Card, S. K., and Mackinlay, J.D. (1989). The Cognitive Co-processor for Interactive User Interfaces. *Proceedings of UIST'89, ACM Symposium on User Interface Software and Technology*, pp.10-18.

[Robertson et al., 1993] Robertson, G. G., Card, S. K., and Mackinlay, J.D. (1993). Information Visualisation Using 3D Interactive Animation. *ACM Communications*, 36(4):56-71.

[Roth and Mattis, 1990] Roth, S. F., and Mattis, J. (1990). Data Characterization for Intelligent Graphics Presentation. *Proceedings of CHI'90, ACM Conference on Human Factors in Computing Systems*, New York, EUA, pp. 193-200.

[Russel and Norvig, 2003] Russell, S. and Norvig, P. (2003). Artificial Intelligence: A Modern Approach, 2nd Edition. *Prentice Hall*, 2003.

[Shneiderman, 1992] Shneiderman, B. (1992). Tree Visualization with Tree-Maps: A 2-Dimensional Space Filling Approach *ACM Transactions on Graphics, 11(1)*, pp. 92-99 .

[Shneiderman et al., 1992a] Shneiderman, B., Williamson, C., Ahlberg, C. (1992). Dynamic queries: Database searching by direct manipulation. *Video in ACM Conference on Human Factors in Computing Systmes - CHI 1992 Video Program, in Proceedings of CHI' 92*, Monterey, CA, pp. 669-670.

[Shneiderman 1994] Shneiderman, B. (1994). Dynamic Queries: For Visual Information Seeking. *EEE Software, vol. 11, 6*, pp. 70-77.

[Shneiderman, 2000] Shneiderman, B. (2000). The Limits of Speech Recognition. *In Communications of the ACM, Vol. 43, No. 9*, pp. 63-65.

[Shneiderman 2004] Shneiderman, B. (2004). Information Visuali-sation: Research Frontiers and Business Opportunities. *Course Notes in the 8th International Conference in Information Visualisation*, London, England, UK.

[Siebra, 2006] Siebra, C. (2006). *A Unified Approach to Planning Support in Hierarchical Coalitions.* PhD Thesis, The University of Edinburgh.

[Siebra and Tate, 2003] Siebra, C. and Tate, A. (2003). I-Rescue: A Coalition Based System to Support Disaster Relief Operations. *Proceedings of the Third International Conference in Artificial Intelligence and Applications*, Benalmadena, Spain.

[Smith and Lassila, 1994] Smith, S. F. and Lassila, O. (1994). Toward the Development of Mixed-Initiative Scheduling Systems. *In ARPA-Rome Laboratory Planning Initiative Workshop*, Tucson, AZ.

[Soanes and Stevenson, 2004] Soanes, C., Stevenson, A. (199). Oxford Dictionary of Englishs. *Oxford University Press; 2nd edition.*, August 1, 2004.

[Sun Microsystems, 2003] Sun Microsystems. (2003). Java 2 Micro Edition (J2ME). `Url: http://java.sun.com/j2me/`.

[Sun Microsystems, 2006] Sun Microsystems. (2006). Java Technology. `Url: http://java.sun.com/`.

[Tate, 1977] Tate, A. (1977). Generating Project Networks. *In the Proceedings of the Fifth International Joint Conference on Artificial Intelligence (IJCAI-77)*, Boston, Massachusetts, USA, pp. 888-893.

[Tate, 2000] Tate, A. (2000). Intelligible AI Planning. *In Research and Development in Intelligent Systems XVII, Proceedings of ES2000, The Twentieth British Computer Society Special Group on Expert Systems International Conference on Knowledge Based Systems and Applied Artificial Intelligence*, December 2000, Springer, Cambridge, UK, pp. 3-16.

[Tate, 2000] Tate, A. (2000). <I-N-OVA> and <I-N-CA> - Representing Plans and other Synthesized Artifacts as a Set of Constraints. *AAAI-2000 Workshop on Representational Issues for Real-World Planning Systems, at the National Conference of the American Association of Artificial Intelligence (AAAI-2000)*, Austin, Texas, USA, August 2000.

[Tate, 2001] Tate, A. 2001. (2001). I-X and <I-N-C-A>: an Architecture and Related Ontology for Mixed-Initiative Synthesis Tasks. *In Proceedings of the Workshop on Planning Scheduling and Configuration Design*, Vienna, Austria.

[Tate et al., 1998] Tate, A., Dalton, J. and Levine, J. (1998). Generation of Multiple Qualitatively Different Plan Options. *In Proceedings of the Fourth International Conference on Artificial Intelligence Planning Systems (AIPS-98)*, Pittsburgh PA, USA.

[Tate et al., 2000] Tate, A., Dalton, J. and Levine, J. (2000). O-Plan: a Web-based AI Planning Agent. *AAAI-2000 Intelligent Systems Demonstrator, in Proceedings of the National Conference of the American Association of Artificial Intelligence (AAAI-2000)*, Austin, Texas, USA, August 2000.

[Tate et al., 2002] Tate, A., Dalton, J. and Stader, J. (2002). I-P^2 - Intelligent Process Panels to Support Coalition Operations. *Proceedings of the Second International Conference on Knowledge Systems for Coalition Operations*, Toulouse, France, pp.184-190.

[Tate et al., 2003] Tate, A., Levine, J., Dalton, J. and Nixon, A. (2003). Task Achieving Agents on the World Wide Web. *In Spinning the Semantic Web*, Fensel, D., Hendler, J., Liebermann, H. and Wahlster, W. (eds.), Chapter 15, pp. 431-458, MIT Press, 2003.

[Tate and Dalton, 2003] Tate, A. and Dalton, J. (2003). O-Plan: a Common Lisp Planning Web Service. *In Proceedings of the International Lisp Conference 2003*, invited paper, October 12-25, 2003, New York, NY, USA, October 12-15, 2003.

[Tate and Drabble, 1995] Tate, A. and Drabble, B. (1995). O-Plan's PlanWorld Viewers. *In the Proceedings of the Fourteenth UK Special Interest Group on Planning and Scheduling*, Wivenhoe House Conference Centre, Essex University, England, UK.

[Telea et al., 2003] Telea, A., Frasincar, F. and Houben, G. (2003). Visualisation of RDF(S)-based Information. *In Proceedings of the Seventh International Conference on Information Visualization - IV*

[The Wearable Group, 2003] The Wearable Group. (2003) The Wearable Group at Carnegie Mellon. `http://www.wearablegroup.org/`.

[Thiagarajan and Gao, 2002] Thiagarajan, P. R., Gao, G. R. (2002). Visualizing Biosequence data using Texture Mapping. *IEEE Symposium on Information Visualization 2002(InfoVis 2002)*, October 28-29, 2002, Boston, Massachusetts.

[Tufte, 1983] Tufte, E. R. (1983). The Visual Display of Quantitative Information. *Cheshire, CT: Graphics Press.*

[Tukey, 1977] Tukey, J. W. (1977). Exploratory Data Analysis. *Reading, MA: Addison-Wesley.*

[Veloso at al., 1995] Veloso, M., Carbonell, J., Perez, A., Borrajo, D., Fink, E. and Blythe, J. (1995). Integrating Planning and Learning: The PRODIGY Architecture. *Journal of Theoretical and Experimental Artificial Intelligence, 7(1).*

[Veloso et al., 1998] Veloso, M., Cox, M., Mulvehill, A., Christey, S. (1998). ForMAT–PRODIGY/Analogy TIE. `http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/prodigy/Web/Mixed-init/`.

[Vision Group , 2003] Vision Group. (2003) Vision Group at Microsoft Research. `http://research.microsoft.com/easyliving`.

[Vision Interface, 2003] Vision Interface. (2003) Vision Interface Group at MIT. `http://www.ai.mit.edu/projects/vip`.

[W3 Consortium, 2004a] W3 Consortium. (2004). CC/PP Information Page. `http://www.w3.org/Mobile/CCPP/`.

[W3 Consortium, 2004b] W3 Consortium. (2004). SWRL: A Semantic Web Rule Language Combining OWL and RuleML. `http://www.w3.org/Submission/SWRL/`.

[W3 Consortium, 2005a] W3 Consortium. (2005). W3C Semantic Web Activity. `http://www.w3.org/2001/sw/`.

[W3 Consortium, 2005b] W3 Consortium. (2005). Resource Description Framework (RDF) / W3C Semantic Web Activity. `http://www.w3.org/RDF/` .

[W3 Consortium, 2005c] W3 Consortium. (2005). Extensible Markup Language (XML). `http://www.w3.org/XML/` .

[W3 Consortium, 2005d] W3 Consortium. (2005). World Wide Web Consortium (W3C). `http://www.w3.org/` .

[W3 Consortium, 2005e] W3 Consortium. (2005). Web Ontology Language (OWL). `http://www.w3.org/2004/OWL/` .

[W3 Consortium, 2005e] W3 Consortium. (2005). Web Ontology Language (OWL). `http://www.w3.org/2004/OWL/` .

[Weld et al., 2003] Weld et al. (2003). CC/PP Automatically Personalizing User Interfaces. `In Proceedings of IJCAI-03.`.

[Wilkins, 1997] Wilkins, D. E. (1997). Human-Computer Interface in SIPE-2. `http://www.ai.sri.com/ sipe/gui.html`.

[Wilkins, 1999] Wilkins, D. E. (1999). Using the SIPE-2 Planning System: A Manual for Version 6.1. *Technical Report*, SRI International Artificial Intelligence Center, Menlo Park, CA, USA.

[Wilkins, 2000] Wilkins, D. E. (2000). ARPI's MAPViS Demonstration. `http://www.ai.sri.com/ wilkins/mpa/mapvis.html`.

[Wilkins and desJardins, 2001] Wilkins, D. and desJar-dins, M. (2001). A call for knowledge-based planning. *AI Magazine, vol. 22, no. 1, Winter 2001*.

[Wilkins and Myers, 1995] Wilkins, D. E. and Myers, K. L. (1995). A Common Knowledge Representation for Plan Generation and Reactive Execution. *Journal of Logic and Computation*, vol. 5, number 6, pp. 731–761.

[Wilkins and Myers, 1998] Wilkins, D. E. and Myers, K. (1998). A Multiagent Planning Architecture. *International Conference on AI Planning Systems.*

[Yang et al., 2002] Yang, J., Ward, J. O., Rundensteiner, E. A. (2002). InterRing: An Interactive Tool for Visually Navigating and Manipulating Hierarchical Structures. *IEEE Symposium on Information Visualization 2002 (InfoVis 2002).*, pp 77-84.

[Yates et al., 2003] Yates, A., Etzioni, O., and Weld, D. S. (2003). Towards conversational human-computer interaction - A Reliable Natural Language Interface to Household Appliances. *In Proceedings of the Intelligent User Interfaces 2003 (IUI-03)*.

[Zadeh, 1968] Zadeh, L. (1968). Fuzzy algorithms. *Information and Control*, 5:94-102.

[Zhou and Hansen, 2005] Zhou, R. and Hansen, E. A. (2005). Beam-Stack Search: Integrating Backtracking with Beam Search *In Proceedings of the Fifteenth International Conference on Automated Planning and Scheduling*, Whistler, British Columbia, Canada.

[Zhou and Houck, 2002] Zhou, Michelle X., Houck, K. (2002). A Semantic Approach to the Dynamic Design of Interaction Controls in Conversational Systems. *In Proceedings of the Intelligent User Interfaces IUI 2002.*

[Zupan and Cheng, 1998] Zupan, B. and Cheng, A. (1998). Optimization in of Rule-Based Systems Using State Space Graphs. *IEEE Transactions on Knowledge and Data Engineering*, 10(2):238-254.