Report on State of the Art Agent modelling in Human Interfaces (with respect to functionality levels).

I-MASS

IST PROGRAMME 20878
### Abstract:
The Report on State of the Art Agent modelling in Human Interfaces with respect to functionality levels represents the step of WP6 “R&D User modelling and interaction”. It aims to provide a framework for modelling system behaviour so as to support users’ activities in the real context. Together with the coming results from further activities analysis, the content of this document will be the basis for designing a series of maps of activities to elicit system requirement supporting users’ research needs.

### Keywords list:
- Activity modelling
- Interface agency
- User model
- Domain model
- Interaction model
- Scenario based design
- Functional requirements
Table of Contents

INTRODUCTION .................................................................................................................. 5

1 SOME PREMISES ............................................................................................................. 6
  1.1 Defining the concept of interaction model from a users’ perspective .......... 6
  1.2 Defining the concept of interface agent ................................................................. 7

2 USER MODELLING IN THE DESIGN OF HUMAN INTERFACES: THEORIES AND
   PRINCIPLES .................................................................................................................. 9
  2.1 User-Agent interaction: the control issue ................................................................. 9
  2.2 Adaptivity to the users ............................................................................................. 10
     2.2.1 Constructing user models in interaction systems: an overview of main
          techniques ................................................................. 11
     2.2.2 Intelligent user interface: practical applications of user modelling theoretical
          principles ................................................................. 16
  2.3 Towards a shift in user interface design principles ................................................... 19

3 AGENT MODELLING AND TECHNIQUES FOR USER INTERFACES .............. 21
  3.1 MAS modelling for Human interfaces (State-of-the-art) ......................................... 21
     3.1.1 Agent-based interfaces .................................................................................... 21
     3.1.2 Interfaces and ontologies ................................................................................ 23
  3.2 User interface design: a semiotic perspective ............................................................ 23
     3.2.1 Supporting user’s control on agent behaviour .................................................. 24
     3.2.2 Agent roles and interaction descriptions ......................................................... 24
     3.2.3 Applying semiotics to Agent communication .................................................. 26

4 TOWARDS THE I-MASS INTERACTION FRAMEWORK ..................................... 32
  4.1 Modelling the activity in context (a new perspective) .............................................. 32
     4.1.1 Distributed Cognition and Activity Theory ...................................................... 32
  4.2 Functionality in the I-Mass system .......................................................................... 34
     4.2.1 Introduction ..................................................................................................... 34
     4.2.2 I-Mass as a tool to access the cultural heritage domain .................................. 35
     4.2.3 Non-intelligent functions in I-Mass ................................................................. 36
     4.2.4 Intelligent functions in I-Mass ........................................................................ 36
     4.2.5 A retrospective of I-Mass functions ............................................................... 38
  4.3 Sketching I-MASS Human-MAS interaction (scenario-based interaction models) .... 39
     4.3.1 Research activity: an overview of relevant Aspects, Tools and Places ............ 39
     4.3.2 Research as a distributed activity .................................................................... 39
     4.3.2.1 Expertise ..................................................................................................... 40
     4.3.2.2 Personal strategies and best practices .......................................................... 40
     4.3.3 Scenario-based design ...................................................................................... 41
     4.3.4 Macrosenario: the “point of view” in Virgilio’s work ....................................... 42

5 CONCLUSION .................................................................................................................. 46
6. REFERENCES........................................................................................................49

7. SUGGESTED BIBLIOGRAPHY .....................................................................55
   Cognition ...........................................................................................................55
   User modelling - web interfaces .................................................................55
   Agents .............................................................................................................56
INTRODUCTION

The Report on State of the Art Agent modelling in Human Interfaces with respect to functionality levels is a public report of WP6 of the I-Mass project. A first step of WP6 “R&D User modelling and interaction”, it aims to provide a framework for modelling system behaviour so as to support users’ activities in the real context.

In order to define at best the parameters and methods for modelling human-MAS interaction in the development of the I-Mass system, we must consider the several levels of system design and development which are likely to be affected by the way users’ models are defined and described. Indeed, designing user interaction does not only affect user interface issues, but rather influences system behaviour at different levels.

In this perspective, we first address the problem of user modelling in the design of human interfaces (theories and principles), then we present agent modelling techniques for User Interfaces. The last part of the deliverable is finally focused on sketching an I-Mass interaction framework.

Particular attention has been devoted to describe the iterative methodology applied in order to get data from the activity analysis in context as the basis of the scenario-based interaction model that strongly will inform the design and implementation of the I-Mass functionality levels.
1 SOME PREMISES

The following two premises are instrumental at establishing the two main issues underlying the report on state of the art on agent modelling in human interfaces, namely: I) It is an open issue whether the desktop metaphor based on task-centered design is still the most suitable for agent based interfaces. II) Within i-Mass the knowledge landscape is the normative framework for interface agents.

1.1 Defining the concept of interaction model from a users’ perspective

As information systems become more widespread the need for a precise idea on how the interaction can be designed and supported becomes increasingly important. Naturally, the increased emphasis on interactive aspects between the users and the systems has lead to a greater interest in the study of human computer interaction. So far the more important improvement in human-computer interaction has been the Graphic User Interface with direct manipulation of objects introduced by the David Liddle’s design team at Xerox at the end of the ’70 for the “personal computer”. But in the last years digital information appliances are popping up everywhere around us, in myriad forms: at the office, on the car dashboard, in our pockets, on and inside our bodies. We are aware that this plethora of processors is becoming interconnected through fixed and wireless networks. These networks are transforming the nature of software: data and applications are becoming increasingly distributed and dynamic. This places new types of requirements on HCI. The Internet is turbulent, extremely complex, and constantly changing. It may be a bit unsettling to think of this as a development environment when such a low level of reliability has been accomplished with the desktop operating system! The tangled Net is a tremendous challenge for both the designer and the user. Current program-to-program and program-to-user interfaces have not been built to deal with such distributed and dynamic environment. The Large knowledge-spaces available to a user create need for ways of distributing control and processing. Agent technology has been one of the attempts to provide an answer to this emerging need.

Agents technology evolved so far as:

- an extension of the software object design model
- an evolution of user automation of tasks by macros and scripting
- an advancement of message filtering and meeting scheduling

However this technology has, in many cases, brought human-computer interaction back to command line and form filling interaction. In other cases it has tried to rest on the idea of user model so to avoid the need to filling in forms at every step of the interaction. Thus, a rethinking of user interaction model is needed if we want to cope properly with the complexities of distributed computing on large information-spaces. If this will happen by new interaction concepts within the desktop metaphor or by reconsidering the desktop metaphor as a whole is an open issue.

As a possible heuristic contribution this deliverable will suggest a shift from task centered interface design (the approach adopted by the Xerox team to design GUI) to activity centered interface design. In the section 4.3 it will be provided a rationale for this shift while defining the concept of interface agents.
1.2 Defining the concept of interface agent

In the I-MASS project several interface agents are fulfilling tasks to achieve the common goal: helping and assisting the user the best they can. A scalable and dynamic organisation structure of agents is available to the user. The agents have to adapt themselves over time to its user’s preferences and habits. The know-how of the agents among different users in I-MASS must be shared. The agent must be able to observe and monitor the actions taken by the user in the interface, learn and suggest better ways of doing tasks. As introduced in (Braspenning et al., 2001) the heart of the I-MASS system uses a Knowledge Landscape (KL), which is the high-level representation of (part of) the knowledge that can be found in the cultural heritage information sources and in the reference works accessible from the Virtual Reference Room. This KL has to be persistent to the user too. Different components of agents comprise the adaptive interface. The organisation of this interface agency consists of information retrieval agents, user agents, assistant agents, KL-user interface agents, user query agents and interface agents. This agency has the responsibility to contextualise the user intention and to provide help in specifying the queries. The first responsibility applies as well to the information-seeking user as the information/knowledge providing user. The providing user is the expert in the field and ‘constructs’ the knowledge landscape and the ontology for the agents. The KL-user-interface agent is the responsible agent for representing the interface components for adding new knowledge to the user. The user agent is the ‘link’ between the knowledge landscape and the user. The user’s activity model defines the attributes of each agent. Examples of different components provided by the interface agents are: tools to access the knowledge atlas; tools to make the user aware of which reference work is being used and presenting tools for the knowledge landscape.

Figure 1-2: A possible interface agency.
Figure 1.2 illustrates a possible interface agency. The figure presents the user group divided in regular users and knowledge providing users. Knowledge providing users are explained more further in this document but are responsible for building the Knowledge Landscape. The user-interface agent is responsible to provide a suitable compositional interface to the user. The user agent and the query agent are not directly ‘visible’ to the user, but are agents working in the background. Therefore they are depicted in a layered approach. (See the rectangles with dotted lines).

The interface agency can be divided in two groups of agents: persistent and non-persistent agents. Non-persistent agents reside on the local host pc’s for instance to keep the session-states of the user behaviour (one user) while the persistent agents save and update their profiles and template-like structures to a user-group profile. The first simple user-group profiles for the group of knowledge providing users could consist of activity descriptions for providers from the libraries and another for providers for museums.

The modelling of the interface agents, which are situated at different levels for the user, must be done by taking into account different kinds of human activities. For some agents the user has some more control over its behaviour then others agents. Different kind of models needs to be constructed to support the user in maintaining and monitoring the agent interaction in a direct and in an indirect way. The query agent in figure 1.2 for instance adapts (indirectly) to the regular user (indirect) but can be maintained by a domain model expert user to operate correctly over the Knowledge Landscape (direct).

---

1 A more definitive framework will be described in D6.4.1 or work package 5.
2 USER MODELLING IN THE DESIGN OF HUMAN INTERFACES: THEORIES AND PRINCIPLES

Agents have been described as powerful tools in mediating communication, both by helping humans communicating with programs, and by supporting programs communicate with each other (Engestrom, 1999).

Even though agents are considered intelligent software, their adoption rises two kinds of issues, the first being related to the organization and filtering of information, and the second concerning the need to design systems that don’t take away user’s autonomy, in other words, that always require intelligent human input in order to take decisions on user’s behalf.

In this chapter main problems in human-agent interaction at interface level will be treated. In the chapter four a distributed cognition approach will be presented as a new way, not yet exhaustively applied to this field, to build activity models for defining user interaction needs over agency in order to augment people performance in decision making, concerning information browsing.

2.1 User-Agent interaction: the control issue

An intelligent agent is a computational software entity that has been developed to help and assist users in the achievement of their goals in a given situation.

Patty Maes (1997) gives a clear explanation for the need for agent technology and this need is related to the current computer environment becoming more and more complicated, and the consequent increasing number of tasks users have to undertake.

A computer today is an open window on a network of information and other people, a continuously changing network. “As we know from other domains, whenever a workload or information load gets to high, there is a point where a person needs to delegate[...] the same will happen with our computer environments: that they become just so complex and we use them for so many different things that we need to be able to delegate” (Maes, 1997).

To be helpful to a user, an agent is required to have:

- a certain knowledge about rules to perform a particular task;
- a certain level of knowledge about user’s preferences;
- some defined skills, in term of the ability of an agent to achieve specific tasks on the base of its peculiar properties (flexibility, adaptivity, continuity, autonomy, learnability, reactivity, inferencing, personality, mobility, collaborativity, communicability);
- and both a user and an application interface (Janca, 1998).

What is actually required to agent technology is to show adaptive behaviour in order to perform tasks in user’s behalf.

Nevertheless the concept of adaptability violates one of the most consolidated principles of control: the consistency. The violation of the consistency principle allows both to take into account the evolution of user’s preferences and the content changes that occur according to users’ actions. The true challenge, Maes states, lies in designing the right user-agent interface, in particular addressing the issues of understanding and control. It seems indeed essential that people feel in control of their lives and surroundings. And, when automata do tasks for them, humans should be comfortable with the actions, in part through a feeling of understanding, in part through confidence in the systems.

In this sense, it can be useful to provide a model of the system that assures users understanding that the system can be controlled, not only indicating what aspects of
the system can be controlled, but providing an obvious representation and set of methods for exercising this control.

One possibility to manage the technical aspect is to devise a computational structure that guarantees that from the technical standpoint all is under control, while the social part of acceptability is to provide reassurance that all is working according to plan. A way to do this is through developing an appropriate conceptual model of the actions, in which the actions of agents are understood in context.

In this way, users can feel more comfortable in their ability to find out what actions have been taken in their behalf, that private matters remain private, expensive or unwanted actions will not be taken without explicit permission, and it will be always possible to trace back the actual sequence of acts.

The idea is to build computer surrogates that possess a body of knowledge both about the thing (a process, a field of interest, a way of doing) and the user in relation to that thing (tastes, inclinations, knowledge level), to let the user the option to do something just when he/she wants to, not because he/she has to.

Because of agents’ promise to hide complexity, to perform actions users could/would not do themselves, then one of people’s attitudes about agent is to have over-exaggerated expectations. People will be more likely to accept intelligent agents if their expectations were consistent with reality. One of the way through which this could be achieved by presenting an appropriate conceptual model, a “system image”, that accurately depicts the capabilities and actions (Norman, 1997).

At the light of what has been said above, design has to deal with many issues. First of all it is necessary to think about the way in which user instructs and controls the agent (i.e. input mechanisms and preferences elicitation and so on…). Secondly the modalities and the nature of the feedback provided by agents has to be addressed. Thirdly the communication between users and agents, as well as advises and search suggestions (e.g. new fields of exploration, new criteria for information filtering, unexpected points of view for researching activities…) have to be considered in the interface design process.

Software agent technology provides solutions to support the users with extra eyes, ears, hands and brains, but in order to make this possibility effective there is a strong need for considering the context in which humans perform their activity. (This aspect will be deepened in chapter four).

Not taking into account the role played by human activities in their whole historical and social contexts (Vygotsky, 1978) can conduct to develop non adaptable systems, that instead of enhancing human performance cause problems in user interaction with information.

In relation to this, particular attention has to be devoted to adaptivity as the main mechanisms by which interactions between human and software agents take place, particularly when talking about multi-agent systems.

2.2 Adaptivity to the users

Adaptivity is a way to manage complexity raised by information growth. It represents a twofold solution since on the one hand it addresses the problem of the information filtering in relation to a specific domain and its context, on the other hand it relates to users’ previous knowledge, goals, background and experience and finally preferences. In this section attention will be focused on the user side of the coin, to explore system adaptivity and proactive behaviour for enhancing users’ performances can be shaped starting from user expectations.
Adaptive systems generally rely upon *user models* to adaptively shape their behaviour: they collect data about the user (his/her knowledge level, goals, and preferences) and then process them in order to create a *model* on which to perform an adaptation process.

![Diagram](image)

**Figure 2.1: Classic loop “user modelling-adaptation” in adaptive system (in Brusilovsky, 1996)**

In traditional software design methodologies, as well as in much of the Human Computer Interaction literature, by ‘user model’ it is intended the designer’s model of the users of a system who are modelled a priori and whose models are built implicitly into the system as it is designed. These models are static and cannot be updated automatically by other components of the system.

An alternative use of the term coined within the framework of Artificial Intelligence treats user models as a explicitly represented data structures, or knowledge sources, often held in a separate user modelling component of the system. In this second meaning (which is the one adopted in this paper) the information in the user model is a representation of the knowledge and preferences which the system believes that a user possesses (Benyon et al., 1993).

In natural language dialog system, as well as artificial intelligent dialog systems in general, a user model thus is a knowledge source within the system, which contains explicit assumptions on all aspects of the user that may be relevant to the interactive behaviour of the system. A user model is composed by different components in the system that are functional to incrementally construct a user model: to store, update and delete entries, to maintain the consistency of the model, to supply other components of the system with assumptions about the user.

### 2.2.1 Constructing user models in interaction systems: an overview of main techniques

Information in the user model may be relatively static and long-term, or it can also be dynamically updated as the user interacts with the system, to enable it to adjust its functionality and interface according to the needs of the single user. Accordingly we can distinguish two main research direction: on one hand there are stereotyping techniques for describing user profiles, while on the other hand there are inference based adaptive mechanisms.
In both cases the main issue to address is how to include all the characteristics of the user that have an effect on the adaptation while excluding irrelevant piece of information.

2.2.1.1 User stereotypes

The notion of stereotype as set of characteristics shared by many users was first introduced by Rich (1989) within the GRUNDY system project where the idea of stereotypes as collections of data which typify class of users was pioneered. Stereotypes can be then defined as a means of making a large number of assumptions about a user based upon only a small number of observations. They have been widely used in user modelling, but their construction has been almost exclusively manual. Attempts to automate the acquisition of stereotypes have been limited to the adaptive refinement of numeric parameters, rather than the actual construction of stereotypes (Rich, 1983). The manual construction process usually involves the classification of users by an expert and/or the analysis of data relating to the interests of individual users. Acquiring the stereotypes in this way is a difficult task, therefore big effort has been spent for forming assumptions on the basis of user’s input into the system, which are the most direct and the most reliable in problem-solving domain. These assumptions of belief or goal can be expressed directly by the user or indirectly inferred by the system. Such inferences can be structure or content based, deriving from the form or the meaning of an input, used to describe mutual beliefs, plans, preferences, requirements and criteria of assessment in a certain domain.

In most systems, stereotypes are applied as tools supporting the system in pursuing a particular set of interactive goals or in behaving cooperatively when performing a task together with the user. It is especially important in dialog systems used by a heterogeneous group of users and characterized by some flexibility in what they tell to the users. These systems act on the basis of the particular user constructed by the system in the course of the dialog. In such cases, stereotypes are used as collections of frequently recurring characteristics of user. Thus, while the user-provided information relevant for the procedure is stored as factual knowledge in the user model, the stereotypes triggered by these facts are stored as hypotheses, which can be withdrawn if they are disconfirmed by the subsequent interaction.

Special cases of the exploitation of user models are represented by their use in various forms of anticipation loops, based on the implicit assumption that the system’s procedures for language analysis and interpretation are similar to those of the user. Furthermore, other techniques have been developed for recognizing misconceptions underlying unsuitable dialog contributions of the user and for responding to these dialog contributions in an appropriate way.

In flexible systems however, it will become necessary that the assumptions about the user inferred are explicitly represented in the user model, and the handling of user misconceptions are performed on the basis of such an explicit model (Kosba et al., 1989).

In order to overcome the difficulty to model relevant information about the user, numerous attempts have been done by researchers to establish the guidelines to design learning agents which detects patterns of user’s action and learn from other agents.

A peculiar technique for enhancing system inferences as concern users’ preferences is based on user plan recognition, which will be presented hereafter.
2.2.1.2 User plan recognition

Plan recognition basic idea is that the system should observe the user’s actions and interpret them in term of the user’s possible plan. On the basis of this information, the system should be able to provide more intelligent and more co-operative advice and assistance to the user.

Early work in plan recognition was associated with natural language understanding, based on the observation that people do not usually make their intentions totally explicit in their utterances, and thus was directed towards mechanisms for discovering these intentions behind utterances, using this information to provide helpful responses (Allen 1983). Plan recognition is particularly useful for intelligent interface technology when context-sensitive help following a particular input from the user cannot be provided using pre-stored templates, but has to be dynamically constructed on the basis of reasoning about the user’s actions in relation to the current system state and the system’s estimation of the user’s current goals.

Another problem of plan recognition concerns how to determine from the evidence available what the user is trying to do at a particular point in time, and then what sort of advice to offer. In this case the intelligence is based on particular user models that capture the uncertain relationships between a user’s goals and needs, observations about the current program state, a representation of sequences of actions over time, and the words in a user’s query (Horwitz, 1997).

User models constitute an indispensable prerequisite for any flexible dialog in a wider domain, not just sweeten-up a dialog system to render it more cooperative. They are necessary for the complex inference process a system activate in order to respond in a cooperative way to the users interactions. Systems must discover the presumable plans underlying the user’s question, represent them into its knowledge base, examine them for hidden obstacles, and provide information to enable the user to overcome these obstacles.

Coming back to the idea of implementing system adaptivity over user models, while guaranteeing user’s control over system behaviour, in recent years many authors (Maes 1997, Schneiderman, 1998) have underlined the need for users to be allowed to inspect and modify their user models in the system. Even if the user model is not in a form that the user can interpret by defined parameters and rules, the user should always be able to predict the effects of the adaptativity, or at least he/she should be able to turn the model off if he/she is not able to fine tune it. It would be better to have some ways by which users can control and eventually alter the model, e.g. providing explanations fitting with their knowledge instead of letting the system guess something of user characteristics and be adaptive.

According to Maes (1994) and Bevan (1996) in particular users should be able to verify the model the system has created to adapt to their characteristics, they also have to understand the how the system operates on the model it has built, in relation to this finally user should be able to control the degree of automation and intrusiveness of the system in respect of their activities. Both Maes and Bevan point out that users work best with gradual automation, and progressive proactiveness of the system.

Besides user modelling methods, literature on the fields describes adaptivity as an element that can occur at different levels of the system. Browne et al. (1999) report a rudimentary rule-based adaptive mechanism that enables the system to produce a change in the output in response to a change in the input.
A simple adaptive system can be enhanced if it has a dialogue record, which allows us to keep a history of the interaction and build inferences over the interaction between users and the system through time. In such cases agents are characterized with a very little knowledge and they become over time more experienced, gradually building up a relationship of understanding and trust with the user, becoming customized to individual organizational preferences and habits.

More sophisticated systems monitor the effect on a interaction model so that possible adaptations can be tried out in theory and evaluated. Such systems must be able to abstract from the dialogue record to capture an intentional interpretation of the interaction, and must include a representation of their own purpose in the domain model. At another level of complexity, systems may be capable of changing these representations and thus reasoning about the interaction.

Other kinds of adaptive systems (Benyon et al., 2000) include a user model in which individual users are assigned to stereotypes based on the use they would make of the data. The intentional level of the domain model represents knowledge such as “for forecasting a graph is preferred to a table, whereas for complex static comparisons a bar chart should be used”. The conceptual level includes rules for selecting the type of chart and the best way to design it. At last, at the physical level the system exploits the preferences of users and employs standard conventions for deciding how to display the data.

Adaptive agents should be able to modify their behaviour according to both other agents’ behaviour, and the history of interaction between the overall system and the user.

Maes (1994) indeed states that in order to be called adaptive software agents should be able to acquire their competence from different sources such as: looking “over the shoulder” of the a user while he is performing some actions; learning through direct and indirect feedback from the users; learning from user-supplied examples; and asking devices from other users’ agents that may have more experience with the same task. This way to get adaptivity requires less work from application developer and helps in transferring information and know-how among the different users of a community. Nevertheless automatic user modelling in adaptive systems does not provide completely reliable solution.

In general adaptive functionality lets systems operate by progressive steps:
- **Noticing**: trying to detect potentially relevant events;
- **Interpreting**: trying to recognize the event (generally mapping the external event into an element in the system vocabulary) by applying a set of recognition rules;
- **Responding**: acting on the interpreted events by using a set of action rules, either by taking some action that affects the user, or by altering their own rules (i.e. learning).

However, (Erickson, 1997) even if such functionality is necessary for enhancing the system, it is often not sufficient, since it may do harm if it confuses the users, interferes with their work practices or has unanticipated effects. The system may fail to notice a relevant event or may mistakenly notice an irrelevant one. It may misinterpret an event that has been noticed. It may respond incorrectly to an event that it has correctly noticed and interpreted (that is it has responding rules that don’t match what the user expects), there’s no guarantee that the user’s interpretation will correspond with the system’s ones. Most models of user knowledge are unreliable because there are many changes in user’s behaviour so that, if the adaptive system assumes a model of user’s plans and goals which is too rigid and static, it cannot capture the continuous variations users are involved in.
Brusilovsky (1996) states that “the only way for the system to get the required information about the user is to involve the user in the process of user modelling and to collaborate with the user in gathering the information”.

Adopting a cooperative approach to user modelling implies an iterative work of user model definition based on cyclic adaptation. In this perspective user-system interaction is a continuous source of information gathering for what concerns users’ needs on which to implement system adaptivity.

This paradigm implies a shift from GUI interaction system approach to users’ data manipulation. Cooperative user modelling indeed is meant to make users manipulate a series of interaction levels which were absent in the GUI interaction model: in the latter the desktop metaphor had been designed to provide the user with the illusion of directly manipulating information items; the cooperative user modelling approach on the contrary assumes that, in order to enhance users’ performance in data manipulation, it is necessary to design proactive systems that act on user’s behalf. In order to perform activities in user’s stead the adaptive system is bound to know how to satisfy user’s goal. Cooperative user modelling is indeed a way to involve the user him/herself in the definition of what his/her expectations are. This implies users interacting with system levels that weaken the direct manipulation paradigm: in order to access what s/he may require in a complex environment characterized by dynamic and distributed information, the user is now asked to specify what he is looking for rather than actually looking for it.

The following figure (Fig. 2.2) may help in understanding the shift from the classical loop “user modelling - adaptation” (see Fig. 2.1) to the “cooperative user modelling-adaptation” vision to better grasp how in the this second proposal users actually interact with the system to allow system adaptation to their needs.

![Figure 2.2: Collaborative user modelling in adaptive systems (Brusilovsky, 1996)](image)

Involving the user in the process of user modelling allows to get additional information from the user and to get subsequently a much more reliable adaptive behavior as regards system pro-activity. Nevertheless, as it has been previously pointed out, involving users into model creation poses an important question in terms of interface design issues. This choice invokes a departure from the classical GUI metaphor: since users are not required to directly manipulate metadata, but rather to collaborate in the definition of metadata what functionalities should new interface provide users?
2.2.2 Intelligent user interface: practical applications of user modelling theoretical principles

The overall framework of intelligent interface technology is based on three different models: domain, user and interaction. The domain model is the representation of the application, at an abstract level, without including details. The user model, as previously depicted, describes what the system knows about the user. The interaction model is an abstraction of the interaction, the dialogue record, along with mechanisms such as a rule-based, a statistical model, a genetic algorithm, for making inferences from other models, for specifying adaptations and, possibly, evaluating the effectiveness of the system’s performance (Benyon et al., 2000).

In this framework it is interesting to notice two main points. One concerns the existence of several user roles and hence several user models which represent the individual agents in a multi-agent system, and similarly more than one domain model. The other refers to the interaction model as expressed through the adaptive, inference, and evaluation mechanisms, which may be extremely complex, embodying theories of language, pedagogy or explanation. While theoretical foundations are necessary to prevent ad hoc solutions that lack generality, practical constraints of performance and usability will constrain the applicability of theoretical research in real systems. Usually, the ideas of plan recognition, user modelling and intelligent help are hard to translate into the actual practice even if at first sight they appear convincing or at least plausible.

Both user modelling and spoken dialogue systems areas involve the implementation of some form of intelligent behaviour in computer systems. Adaptivity is normally considered a relevant indication of intelligence. Applied to computer systems, this implies the ability to adapt output to the level of understanding and interests of individual users. User modelling has been concerned with developing systems that provide such an adaptivity by acquiring information about users such as their goals, plans, preferences, and beliefs, and then using this information to control the system’s output. In Artificial Intelligence many researches, focused on formalisms and techniques to support plan recognition, have been carried out. User plan recognition, as previously mentioned, enables the system to infer the user’s goals and plans using evidence from the user’s input and previous interactions with the system. Another indication of intelligence is the ability to communicate. Almost since the beginning of Artificial Intelligence, there have been attempts to develop computers that could simulate human conversation and spoken dialogue systems, so that spoken dialogue systems are practical outcome of these efforts. In fact, while intelligence in user modelling and spoken dialogue systems is likely to be inspired by theoretical research, in AI, the need to make this intelligence actually work in commercially deployed systems and to meet the requirements of usability and acceptability poses constraints on the theory (McTear, 2000).

In the introduction to the Proceedings of the Sixth International Conference, UM97 (Jameson et al. 1997) there have been listed a variety of purposes fulfilled by user models:

- Helping the user to find information, e.g. when navigating the Web by updating hypertext links and recommendations to the user’s previous navigation behaviour, or filtering the WWW documents in accordance with the user’s interests;
- Tailoring information presentation to the user’s abilities and preferences;
- Adapting an interface to the user, e.g. offering Web navigation shortcuts that reflect past accesses, or providing special support and interface simplifications for novice or disabled users;
- Choosing suitable instructional exercises or interventions;
- Giving the user feedback about their knowledge;
- Supporting collaboration;
- Predicting the user’s future behaviour.

As most research in user modelling during the 1980s was concerned with theoretical issues, it borrowed extensively from AI and used prototypes just to demonstrate these theoretical approaches. However, in the early 1990s researchers began to consider the commercial potential of the technology and to investigate the practicality of some of the earlier theoretical research (Krause et al., 1993). Some of them proposed a minimalist approach to user modelling which provides just enough assistance to a user through a pragmatic adaptive user interface and demonstrated that the minimalist user-modelling component improved the subjective measure of user satisfaction.

2.2.2.1 User modelling for information retrieval on the web

The typical model of a user retrieving information (on the web) is derived from the information retrieval model: the user is his/her query. Thus any user typing the same query gets the same response from the database. The models of information retrieval are based on similarity of the information request and the objects in the information space. Despite its severe limitations, this user model has been successful and has survived thanks to its computational simplicity. Its success is evaluated using a set of queries and determining the ability of the search engine to order documents by their relevance, measuring it by recalls and precision. However, this success is based on the premise that relevance can be determined solely on the basis of the query, and not on the users, their past history, current circumstances, and the future of the retrieved information [Wilkinson, 1999].

There have been important strands of information retrieval research that break down the model to some degree, the mainstream of which is the one associated with relevant feedback. In this environment, after an initial query, the user specifies which of the presented documents are relevant or not. This supplies both a history to be built up and more knowledge about the user requirements, in addiction to simply modifying the query, letting another matching take place.

In TREC [Voorhees et al., 1998] they concentrated on the retrieval of information, where the task is to find different aspects of an answer, so that it is not simply good enough to identify documents as more or less relevant, but it is necessary to find pieces of evidence for each aspect of a topic. In this way they start to see that different users may have different information needs, but it is still the case that there is no attempt to account for individual differences.

Significant research has been done on investigating researchers behaviour in a library environment. A key reason for limited activity in user modelling information retrieval is that it is still not clear what it can be done with a complex model. There are algorithms for efficiently matching sets of queries among large sets of documents, but if the user model is represented as a set of constraints, dependencies, and a complex history, there are no methods of matching information needs. According to this it is necessary to build user model that are enough representative to allow individual information needs to be guaranteed, and yet sufficiently simple to make the matching possible.
Several components, which reflect user needs in a more accurately way, are necessary for retrieval information. One is the recognition of the need of different types of answers from the ones not desired of the list of documents found. One way of discovering different aspects of a topic may be to develop cluster of information in order to let the users build their own map of the information space. In a different situation, it is also possible that users may not so much be seeking specific answers as seeking points to start exploration in an information space. Anyway it is important to model the nature of the user’s desired answer.

Another component that makes the retrieval successful is to recognize that there may be no single document providing the information needed. Thus, parts of different documents, and information from different databases may need to be synthesized into a virtual document, created properly to satisfy a specific user’s need at the time.

A further factor is the acknowledgement of the continuous user model changing as the retrieval process takes place. This occurs iteratively during a dialogue: as any points are reached during the interaction, some information is no longer relevant and other information becomes important. In this context it is important to catch how discourse and dialogue models influence user models during an interaction [Wilkinson, 1999].

Coming to pragmatics, nowadays it is hard to implement sophisticated user models in a high volume environment such as the web. There are several key elements of a user model to be analysed which lead to significant gain in the success of a user’s interaction with a retrieval system. Some of them are:
- identification of the user to allow history to be taken into account;
- identification of the type of answer needed to allow one of a variety of types of answers to be selected;
- identification of the preferred language of the user to allow appropriate delivery;
- identification of the volume of information needed to allow tailoring and relevant abstracting to take place.

As stated in paragraph 2.1.1, Adaptivity to the users, reliable models can be built in cooperation with the users themselves.
2.3 Towards a shift in user interface design principles

In this chapter we have tried to outline what are the main issues involved in multi-agent based system for information management. We have addressed the problem of system autonomous behaviour in relation to users’ goals, in particular for what concerns system adaptivity to users needs, and we have very much focused on the control issue as concerns the consistency principle violation that follows agent based system behaviour.

These themes imply a need for rethinking to the current main model of addressing to user interaction interfaces, i.e. the direct manipulation of data on the desktop metaphor. As stated in the premises of this document it is still not clear whether the GUI model should be abandoned, but there is an important quest for analysing the consequences of shifting from environments where data could be easily managed through direct manipulation to environments where information is distributed in remote repositories and dynamically updated with overwhelming frequency.

The change currently undergoing can be illustrated by the following figures:

In this first figure it can be seen how users’ information needs could be satisfied by using facilitating users’ access to data and application by providing them by a commonly shared metaphor, namely the desktop metaphor of the Graphic User Interface operation mode. As previously stated GUI allowed user to have the impression to directly manipulate object in the information space.

As accessing to information becomes more and more difficult for the increasing number of available data, information needs to be filtered, in order to orient users towards those items s/he may require to accomplish her/his tasks:

In this second case, in order to face information management complexity, profiling users on one side and restricting access to specific cluster of information on the other
side is only a partially working solution. Indeed, the left and the right cone in many
cases cannot match producing misleading results.

Furthermore, while modelling cognitive processes of users’ categories allows to
restrict users attention and thus control only onto those items which are relevant for
him to perform a peculiar task, “the human is often reduced to being another system
component, with certain characteristics, such as limited attention span, faulty
memory, etc. that need to be factored into the design equation for the overall human-
machine system[. . .]. Individual motivation, membership in a community of workers,
and the importance of the setting in determining human action are just a few of the
issues that are neglected” (Bannon, 1991).

The problem we are addressing when talking about distributed high volume date
repositories, nevertheless, cannot find a solution in user profiling. In this last case
indeed the mail goal is not restricting user control over those functionalities that
enable him/her to perform a given task. The problem here is to enable users capacities
of control within a larger context, which is the context of users activities on the new
domain of interaction introduced by agent technology.

As it will be argued in chapter four there is a strong need for analysing the role of new
artefacts and understanding of the way humans interact with them within the context
of their own environment. Activity theory provides a paradigm for the description
of the role of tools and social interaction in the structure of human activity. The need
for such an approach lies on the assumption that human activity is mediated by tools
in a broad sense and human higher mental functions must be viewed as products of
mediated activity.

As Bannon states: “understanding people as actors in situations, with a set of skills
and shared practices based on work experience with others, requires us to seek new
ways of understanding the relationship between people, technology, work
requirements and organizational constraints in work settings”.

As concerns the problem of interface design and the role of multi-agent systems in
filtering and organizing information according to users activity, goals, knowledge
base, and preferences, it seems clear that studying users in their work context can
provide useful information, overcoming the limits of a cognitive approach. Even
clearer appear the necessity of involving user in the design process is indispensable: it
represents the only way to ensure that the resulting computer system adequately meets
the needs of the users.
3 AGENT MODELLING AND TECHNIQUES FOR USER INTERFACES

This chapter describes several techniques, architectures and related topics regarding the modelling and design for interfaces.

3.1 MAS modelling for Human interfaces (State-of-the-art)

The variety of different kinds and types of interface agents existing nowadays is very broad. They can be classified in direct interface agents and in more indirect (autonomous) agents. Direct interface agents are actively assisting a user in operating an interactive interface: personal assistant agents, avatars, tutoring agents, context-sensitive help system agents, and recommending agents. The more indirect (autonomous) agents are working in the background: information filtering, information discovery agents, adaptive agents, information agents, reactive agents, query processing agents, user model agents. For more classifications and overviews of interface agents we refer to (Wood, 1994), (Brown & Santos, 1999), (Maybury, 1999) and (Nwana, 1996). In (Brown et al., 1998) the authors also consider interface agent requirements. (Brown & Santos, 1999) describes that in order to enable interface agents to interact effectively and efficiently some sort of user model is needed within the system. This user model allows the system to adapt its response to the user’s preferences, biases, expertise level, goals and needs.

Seen from an architectural perspective it is a key issue for a lot of system developers and modellers to find limited models of users (interests) and agents. Especially for mobile agents in lightweight approaches like LEAP (Berger et al., 2001) it is important to keep the architectures and the models minimal. Another requirement for mobile agents is that the user can use the agents with minimal effort.

In this section we make a distinction between two groups of users. The first group that we distinguish are regular users who are browsing the I-MASS system and thus also the Knowledge Landscape. Another group is the expert-users group who provide new knowledge to the core system. For both interface-user groups we describe MAS modelling approaches. The next subsection focuses mainly on the first group and subsection 3.1.2 on the expert-users group.

3.1.1 Agent-based interfaces

This paper mentioned earlier the approach of Benyon (Benyon & Murray, 1993): Benyon’s main models are the domain model, the user model and the interaction model. The interaction model contains the adaptation mechanisms and is situated between the user model and the domain model. This subsection firstly focuses on the adaptive interface agents.

Maes’ Modelling adaptive autonomous agents (Maes, 1994) examines various AI approaches related to agent-technology and compares behavioural and traditional AI-based agent architectures. (Maes,1994) discusses mainly (behaviour-based) agent architectures: “the totality of a set of principles, an organisation and the set of tools, algorithms and techniques that support the construction of autonomous agents, an “architecture” for modelling autonomous agents”. The observation from a behaviour-based point of view is that a complete intelligent system is situated in some environment and that it adapts itself, not relying on internal representations. The two sub-problems involved in modelling adaptive autonomous agents that Maes mainly tackles are relating to the architecture itself. These two sub-problems are the problem of action selection and the problem of learning from experience. Maes’ observation is that most existing architectures have only concentrated on one problem at once. Table
3-1 represents the comparison between Traditional and Behaviour-based AI concerning the modelling in these architectures:

**Table 3-1: Comparison Traditional and Behaviour-based architectures regarding modelling.**

<table>
<thead>
<tr>
<th>Traditional</th>
<th>Behaviour-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decomposed along ‘Functional modules’</td>
<td>Specific task modules</td>
</tr>
<tr>
<td>Agent has a complete correct internal model; an accurate copy of the environment inside the system</td>
<td>No central representation but locally task-oriented modules that are not related</td>
</tr>
<tr>
<td>Different modules responsible for internal representations and are centrally sequentially organised</td>
<td>Highly distributed parallel organisation</td>
</tr>
<tr>
<td>Activity modelled as result of deliberative process</td>
<td>Goal directed activity</td>
</tr>
</tbody>
</table>

Maes also shortly describes how to build adaptive autonomous “interface agents”: “Several competence modules are constructed that are experts (or try to become experts) about a small aspect of the task.” The agent is supposed to learn the user’s behaviour by gathering information and keeping statistics. Several components are learning and are executing commands when new situations appear. The action sequences are provoked by the distributed system. This system adapts to the user behaviour, reacts fast and seems to “understand” the user intentions.

The Collaborative interface agents in (Lashkari et al., 1994) are self-learning interface agents to assist the users dealing with the applications. In the learning period the user has the possibility to provide feedback in three ways: with the help of an activity monitor; with an explanation facility and with an interface to browse and edit the agent’s memory. In (Kozierok & Maes, 1993) the authors have provided the system with explanation facilities of why the agent suggested a particular action.

The interaction model of (Benyon and Murray, 1993) is a vital part of an adaptive system and the developer of these systems has to decide upon the level of abstraction for the dialogue record, the individual user data and the interaction knowledge-base. In Benyon and Murray’s approach the interaction model must be constrained so that all attributes used in the representation are defined in the domain model or in the user model as appropriate. Automatic mechanisms are required for this through software support. (Bylund and Waern, 1997) present the notion of adaptation agents that adapt an open range of services to a common user model. “The adaptation agent is intended to be owned by the user, who can inspect and control its content and decide how information can be distributed to services”. In (Bylund, 1999) Bylund has made a division influenced by Benyon’s information models and switched focus from design aspects of adaptation to the matters of how the adaptation process can be distributed. Both works of Bylund and Waern were conducted within the Kimsac project, Kimsac is a service architecture that makes use of adaptive help. Agents publishing their services is a mechanism to close the gap between the user and the designer of ontologies. (designer has to think about how the agent must maintain a shared domain
model). McDermott is one of the participants in DAML\textsuperscript{2}’s automatic tools for mapping among ontologies. (McDermott et al., 2002) focuses on agents providing service descriptions. Glueing code is coined for the procedure that translates from what the agent knows to what it needs to send. Planning algorithms are provided as solution to cope with disparities between several agents and their service descriptions. Reconciling contradictory ontologies is required by human intervention while the rearrangement of the data structures of message-sending can be automated with agents. Subsection 3.3.3 even claims that with the support of papers of Euzenat (Euzenat, 2000) also the agent interaction protocols can provide support for this first requirement (e.g. human intervention).

3.1.2 Interfaces and ontologies
One of the most recognised requirements for agent-based systems but also in the Semantic Web initiatives is the importance for the support of sharing, exchanging and reusing ontologies. From the point of view of the interface it is its ‘delivery’ task to close the gap between the users of ontologies and the designers of ontologies. The importance of the use of semiotics for the interface by constructing ontologies with DAML+OIL is underlined in (Bechhofer et al, 2001) The paper claims that the modelling approach is lacking and that there is a need for semiotics to reproduce representations on the side of the receiver and given the intent of the sender.
In Frame-Based knowledge representation systems the user, who wants to provide new knowledge, can be guided in entering the new knowledge. Minsky proposed the notion of frames to adopt a more structured approach to connect chunks of knowledge together with facts about objects and procedural knowledge. The “top levels” of a frame are fixed and represent things that are always true about the supposed situation. (Minsky, 1975). The importance of interfaces in a system with a frame-based facility is explained by the following: “When a user creates a new frame and that frame inherits slots from its parent, the inherited slots form a template that guides the user in filling knowledge about the new concept. Because all slot and facet information is available at run time it is accessible to a program such as a user interface that guides the user in entering new knowledge. The user can directly determine the slot data-type and value restrictions.” (Karp, 1993) In I-MASS the agent platform JADE has been chosen and since this framework allows ontologies to be defined using a frame-based language we extended the Knowledge-Landscape-INCA-COM editor to generate JADE ontologies. This INCA-COM editor is also described in D3.5.1 and can be found at (Kruijsen et al., 2002). Related work can be found in (Cranefield & Purvis, 2001).

3.2 User interface design: a semiotic perspective
Goguen coined the term recipient’s design concerning the development of dialogue structures between agents. His work (see URL at (Goguen, 1999)) emphasises the representation of knowledge in relation with the use of this knowledge (and it’s representation). The main emphasis of Goguen is the important factor that when sending a message the receiver also needs the capability to reconstruct the message’ meaning. Since applying semiotics is becoming a new trend in agent communication we will focus on this perspective.

\footnote{Darpa Agent Markup language, http://www.daml.org}
3.2.1 Supporting user’s control on agent behaviour

De Souza shows in his work how Knowledge Representation and Semiotic Engineering can give support to software designers to explain their design rationale to end users. The paper explains a programming experiment with a robot combining HCI, Knowledge Representation and End-user programming. In the proposed environment the authors posed the following requirements:

- The user has to be told about the original programmer’s decisions.
- The user has to be told about the programmer’s/designer’s interpretations and intentions.
- And the user has to be told how he can reconfigure the original model and express his/her own interpretations and intentions.

In fact the robot programmer has to generate the code for the robot behaviour and also the explanation about the code. De Souza concludes that end-user programming has the requirement that users have to learn more about applications than in most HCI scenario’s. But “in semiotic terms, reasoning systems allow users to gain one degree of perspective on a designer’s interpretation of a given domain, its objects and tasks, as well as on his or her encoded abstractions and arbitration in building a computer model of the domain and its implementation”. (de Souza, 1997)

3.2.2 Agent roles and interaction descriptions

Nowadays more attention is given to how to define ontologies that can be used by agents. The paper of (Labrou et al. 1999) claims that “it would be useful to standardize some of the basic conversation protocols for the more fundamental tasks.” And that the availability of these protocols “will reduce the overhead of agent development and will allow for the shift of focus of what agents do”

One of the characteristics of agents is its capability of reactive and pro-active behaviour. A kind of control mechanism is needed when several agent behaviours are in competition, a problem that quickly arises when a system is decomposed in different agent modules. “The design of behaviour-based control systems is the formulation of effective mechanisms for co-ordination of behaviour activities into strategies for rational and coherent behaviour” (Pirjanian, 1999) In these behaviour-based systems, the agents are selecting the most relevant actions. In the same paper the author gives an overview of the state-of-the-art of the issues for action selection mechanisms. But for now we focus on the ‘actions’ of the agents because these selection mechanisms are more described in the theoretical framework.

Action-selection mechanisms are related with the interface and so are the “multi-valued behaviours output relevant because they define an appropriate interface and means of communication between the modules”. (Pirjanian, 2000) The action selection, the interaction descriptions and the acts as described next are related, since they all are part of the behaviour of agents.

The interaction descriptions between agents can best be introduced by explaining FIPA Interaction Protocols. The FIPA has standardised a language for the Agent Communication and called it ACL. (Note that the ACL is elaborately explained in Deliverable 3.5 “Architectural framework”. This language is based on a language philosophy theory called Speech acts. The main idea of the Speech Act Theory (Austin, 1962) is to treat communications as a particular form of actions.

---

3 Deliverable 6.4
Nowadays the ‘speech act’ has become the main ingredient for agents to co-operate: the Knowledge Query and Manipulation Language KQML and FIPA ACL are based on the Speech Act Theory performatives (Searle, 1969). One of the most important derivations from the Speech Act Theory for achieving agent inter-operability is the Communicative Act because it characterises the meaning of exchange between agents and thus can be used as a generic tool. The Communicative Act is a simplified model for agents of what Austin calls the speech acts for humans.

An Interaction Protocol (IP) is a kind of dialogue structure, one of the generic tools that FIPA specifies. IP became a basic term in Agent Technology and are descriptions of interactions in terms of message responses. In their specifications the FIPA provides a set of IP’s to support complex interactions composed by many messages. (Bergenti & Somacher, 2000) Because IP’s represent basically dialogue structures between agents and are put in a stylised form, they are more important than an individual communicative act. These IP’s can be regarded as a kind of dialogue templates and they make the users (agent developers) to follow structures from commitments already made.

Each Interaction Protocol uses more than one performative. A performative is a speech act or a communicative act, in which the utterance itself induces a result. In short a performative is the speaker’s (agent’s) intent of the effect of a message.

“The communicative acts have FIPA-specified formal semantics on the side of the sender of a message. But these semantics don’t impose constraints on the behaviour of the recipient” (Steiner, 1999). Because the communicative acts are not ready to be used within dialogue structures the FIPA (pre)-specified these structures. Implementing these structural standards ensures that the agents automatically follow (some) semantics of these ‘structures’. The Interaction Protocols are recognised as suitable ‘template’-structures so that developers don’t have to plan each speech act explicitly. The structure is a ‘basic conversation’ that is a composition of many messages. In some of these structures several parts can be considered as negotiating constructions. Analysing these (complex) conversations can make manifest repetitive structures. Two examples of communicative acts that can appear in a repetitive structure are (request and cancel) or (propose and reject-proposal). Agent negotiating mechanism within a system can often be compared with the auctions that are used in normal life. The negotiation structures in auctions have been paradigmatic for the FIPA Interaction Protocols. Negotiation is an important characteristic within the system since negotiation of context results in that agents can use abilities to learn more about each other ‘interests’ and preferences.

The FIPA has specified 11 interaction protocols of which 4 can be regarded as generic support for negotiation between agents. The 11 interaction protocols are: Request, Query, Request-When, Brokering, Recruiting, Subscribe and Propose. And the four for ‘negotiation support” are Contract Net, Iterated Contract Net, English Auction and Dutch Auction. In these Interaction Protocols the agent fulfils different roles.

The difference between several potential capability levels of the agents and the use of interaction protocols for the agents can best be explained by focusing on the (meta)-models of Joslyn. The model in figure 3.1 is an extension of the model as appeared in
(Joslyn, 2001). The sensor and the agent are representations of the world. The relation between the sensor and the agent is a newly produced representation through generalisation. The relation is a prediction and in the system it is called a request. This kind of model prediction is a syntactical production. The system takes measurements from its environment, and constructs generalized “beliefs”: stored representations of the current state and memories of past states. This is called symbol grounding: the semantic relation to the environment. It is here where a problem can be identified: this problem stems from an explicit dependence on

1. Pragmatic grounding (situating) of models through some control relation and
2. Presence of interpreting agents as choice-making entities; human, robotic, computational agents. The issue is then to find abilities of models to make predictions about potential actions and to find higher levels to set goals of lower levels. (The model of Joslyn is also used by (Braspenning Rome, 2001)).

Figure 3.1: Modelling as a semiotic relation

In figure 3.1 we find two scenario’s (level 1 and 2) querying agents, which show the distinction between agents’ capabilities to resolve queries. The agents possess knowledge about a particular reference work but the user query (and it’s interaction with the (user) interface agent) defines the decomposition of the query over the agents. In the first scenario an agent is triggered and makes a simple request (see level 1 in the figure) while the second scenario agent is more specialised. If the user triggers the agent on level 2, h/she is in fact making nested queries over the system (level 2). The decomposition of the query in the second level is the request-proposal action. Making the user aware of the difference between the two scenarios improves the system’s use. (Requirement B001 of I-MASS deliverable “User requirements”: the user should be aware of the time needed to accomplish a task)

3.2.3 Applying semiotics to Agent communication

Figure 3.2 shows Euzenats example (Euzenat, 2000 a) which brings us back to Goguen’s recipient design.
Figure 3-2: Do these representations mean the same? (Euzenat, 2000 a)

The example in figure 3-2 can best be explained with the help of Euzenat’s Exmo homepage (Euzenat, 2000b). Euzenat explains on this page the need for semiotics: “For instance, one can express knowledge as a class hierarchy and translate it to an interoperability language for communicating it to someone else. If that language expresses the knowledge under the form of clauses (while preserving the interpretation of the assertions), the sender might have difficulties to recognise this semantically equivalent result. Thus, when knowledge is translated between several representation systems, good understanding cannot be given for granted. The main explanation is that human understanding depends on the form although correctness of computer manipulation only relies on semantics. Hence, a semiotic (or pragmatic, as said in linguistics) treatment must be attempted for easing the reconstruction of meaning. “

Euzenat’s paper (Euzenat, 2000a) claims that a semiotic treatment must be attempted for easing the reconstruction of meaning. The development of an understandable sign system can ensure semiotic-level interoperability instead of ‘syntactic and semantic’ levels ‘alone’. Applying sign interpretation rules for the sender and receiver of messages should ease the interpretation level so that the receiver can interpret the message correctly with her/his own rules. This technique of applying sign interpretation rules is alike those of user profiles applied to language interpretation.

(Ouksel, 1999) describes the framework for SCOPES in which a semiotics framework is proposed for the constructing of interaction context. Figure 3.3 illustrates the architecture of SCOPES agent. This framework provides an extensive blueprint to guide their research in interaction and knowledge networks. For the communication layer of the Scopes agent we can situate the FIPA platforms, at the SCOPES Dialogue layer we can place the FIPA Interaction Protocols. The SCOPES Knowledge wrapper layer services requests from other SCOPES agents. The SCOPES knowledge layers also interact with tools for semantic reconciliation. The Context Negotiation layer of SCOPES contains automatic and semi-automatic techniques. The negotiation layer is used for the contextualisation. Therefore mutual belief is expected as important diagnostic for defining communities of interest. The role of the SCOPES human interface module is to provide mechanisms for human intervention into the semantic reconciliation process. The Social Model layer is an important layer between the Human Interface module and the agent-based negotiation layer of SCOPES.
Goguen introduces in (Goguen, 1999) several problems of and explorations into Algebraic Semiotics, a new approach to user interface design. The approach of Algebraic Semiotics is based on the sign, whereby the system, its syntax, its semantics and the rules of interpretation of the user are most important. The representations (of the knowledge and functionalities in the underlying system) have to be considered in relation with their use. Goguen calls the representations, which appear as mappings between sign systems: morphisms. Morphisms should preserve as much structure as possible. Goguen considers the interface as a representation of the underlying functionality to which it provides access. The quality of the user interface is what it preserves and proposes it as an effective basis for design. The interface and its underlying architecture are sign systems and these systems usually have a classification of signs into certain sorts, and some rules for combining signs of appropriate sorts to get a new sign of another sort. Goguen mentions the need for further work on ‘higher order’ signification by providing some form of ‘meta-spaces’.

Concerning the design of agents and applying semiotics it is worthwhile to mention shortly the work of French and his colleagues. (French et al., 1999) instantiated generic semiotic notions within an Information Systems with the help of the Shared Meanings Design Framework (SMDF) With this framework the aim of the authors is to bridge the ‘gap’ between the semiotic paradigm and interface design. The focus of the authors is mainly on E-Finance web sites but several ideas can be applied for any user interface design. In their belief end user received meaning is dependent on
interrelated semiotic levels that are shown in Figure 3.4. The life cycle model in Table 3.2 completes the vision of the SMDF authors.

**Figure 3.4. End user received meaning dependencies & semiotic layers**

In figure 3.4 the layers represent:

- **Layer-1**
  - *HCI ‘signs’*: icons, symbols, currency formats, textual and iconic cues, states and actions – all comprising surface level complexity of WAP or any interface;

- **Layer-2**
  - *Organisation layer*: workgroup, departmental, branch, structure and culture and value systems – both implicit and explicitly articulated;

- **Layer-3**

Semiotics recognises that for meanings embedded in an interface to be fully shared, it is necessary that the context within which the system is to be used should be analysed. Hereby the full richness within user’s workgroups should be captured. The authors suggest to ask questions to find features of user’s workgroups which can be applied to I-MASS also to model the user’s cultural background as well as various macro and micro organisational factors. The following two questions are for instance also useful for I-MASS: What are informal social relations like (do members of the workgroup often assist each other informally) in order to carry out routine tasks? And do workgroup members commonly use any specialised terminology of their own, etc..

The Life-cycle Model in table 3.2 describes the phases in the system development and the corresponding activities and techniques.
**SYSTEMS DEVELOPMENT** | **ACTIVITIES & TECHNIQUES**
--- | ---
Elicitation phase | Apply semiotic analysis of stakeholders. Check status and validity of shared meanings. (Shared Meanings Elicitation)
Human Computer Interface Design | Undertake detailed semiotic analysis of signs and sign systems employed. (Semiotics of HCI)
Quantify surface level complexity | Use state notation to verify interface complexity in terms of user actions and system states. (State Notation)
Identify Tools | Identify semiotic compatible development tools for prototyping/full implementation of the final system. (Semiotic Compatible Checklist)
User Testing | Carry out user validation of the interface. (Semiotic Enhanced Usability Metrics)
Post-Implementation | Use a *Semiotic Review Checklist* to ensure that the system is potentially maintainable

(Figure 3.4 and table 3.2 are reproduced from French et al., 1999)

**Table 3.2: The life-cycle model of SMDF**

In SMDF the authors also apply semiotic analysis:

“*Semiotic analysis attempts to separate out the following features of an interface:*  
*F1:* the intrinsic meaning inherently present within the symbolic system  
*F2:* the perceived meaning as decoded by a particular set of users  
*F3:* the meaning which a particular context may endow to an interface and a set of users.” (French et al., 1999).  

The life cycle model of SMDF cannot be applied in I-MASS but these three features in the semiotic analysis phase can be combined with the framework of Benyon and its three models (domain, user, interaction). Feature F3 fits in the interaction model of Benyon, feature F2 and F1 fits within the domain model, while F2 can also be placed in Benyon’s user model.  

Figure 3-5 represents the overall architecture for an adaptive system according to (Benyon & Murray, 1993).
Chapter 2 described and introduced these 3 models of an adaptive system. The domain model serves a number of purposes. It is the basis of all predictions and predictions which can be made from the user-system interaction. It defines the adaptive aspects of the system by describing representations of domain features. And it holds application characteristics to effectuate it against the required criteria. This model is suitable for the construction of the KL, since there can be more than one domain model. The user model contains a representation of the knowledge and preferences which the system ‘believes’ that a user possesses. The interaction model is the representation of the actual and designed interaction between user and application. It is possible to constrain the interaction model with the help of automatic mechanisms so that the representations in the domain and user model are appropriate. The knowledge for the user model can be acquired implicitly by making inferences about users from their interaction. But it is the interaction model which is primarily responsible for inferring knowledge about users.

A Computational Semiotics view on user interaction allows for more explicit modeling of the required interaction modes. Hence, also more rational decisions about what can be left implicit and what must be explicit in that interaction. First people’s activities have to be observed to find what is needed for the user and interaction model. These activities are needed for the scenario’s. These scenario’s will be regarded as a specification of the interaction between the user model and the domain model. Pre-scripted interactions with standard dialogue presentations will be a mixed combination of these scenario’s and the Communicative Acts and Interaction Protocols of FIPA.
4 TOWARDS THE I-MASS INTERACTION FRAMEWORK

In this chapter the modelling issue within the I-Mass project is addressed, focusing on end-users’ activities (knowledge providing users’ activities will be treated separately within the works of WP2).

We also provide some early stage results of the activity analysis process and some preliminary conclusions.

Since modelling is performed at different stages of system design process (analysis, design, development) and is related to several levels of the system under development (user/activity modelling, agent modelling, system architecture), the models which are proper of each discipline (which focus on different aspects of the whole I-Mass system) must be consistent and well integrated. To a certain extent, this has been achieved within the present document, which proposes solutions which integrate different perspectives and define a common framework for the design of user interaction with the I-Mass system. On the other hand, the representations of the activity and of user-system interactions that have been developed, both in quasi human-like languages and in more structured formalisms still need to be fine tuned in order to achieve the best integration among the different representations of human and system behaviour at the different levels of the I-Mass architecture.

4.1 Modelling the activity in context (a new perspective)

In the following we will present the theoretical approach proposed within I-Mass to model user activity to support the user-MAS interaction.

4.1.1 Distributed Cognition and Activity Theory

Cognitive psychology has been studying the human thinking with the aim to understand the mental process on which is based the human elaboration of symbolic information. The ability of the human mind in elaborating symbolic information is strongly bounded; carry out a complex reasoning without the aid of tools is very difficult, frequently impossible. Indeed, the most powerful forms of thinking always take place in interaction with tools because their use allows to overcome the processing limitations of the human mind.

The tools representing, storing and processing information have been defined by Norman (1991) cognitive artefacts, i.e. writing, printing, photography, computer….. Cognitive artefacts embody a part of the intellectual story of a culture and are the expression of a theory on the human activity: everyone who use an artefact implicitly accepts the underlying theory although frequently unawares (Resnik, 1987).

The distributed cognition approach claims that the human cognitive activity is not exclusively characterised by the brain activity but, on the contrary, it is distributed among the brain and the cognitive artefacts employed carrying out the activity.

This idea dates back to the sovietic historical-cultural school that asserted that all kind of conscious human activity is structured by the use of external tools. Vygotskij (1974) defined the principle underlying the distribution of the cognitive activity as the principle of the extracortical organisation of the complex mental activities. He claimed that even the most elementary artefact as a notch on a stick or a knot in one’s handkerchief, modifies the structure of the psychological processes. The creation and the use of this kind of tools extend the range of operations of the psychological processes beyond the biological dimension of the human nervous system allowing to incorporate artificial stimuli produced by other persons or auto-produced by the
psychological operations themselves; thus, the incorporation of external representation in the psychological processes allows new forms of behaviour.

The theoretical position of Vygotsky - known as Activity Theory - can be summarised around three main assumptions and related principles:

• Higher mental process in the individual have their origin in social processes
• Mental process can be understood only if we understand the tools and signs that mediate them
• Mental process can be properly understood only considering how and where they occur in growth

4.1.1.1 Methodology

Activity Theory (AT) does not present a clear methodological prescription for the description or analysis of behaviour in anything like the set of procedures used by western experimental psychology. Christiansen (1996, p. 177) summarises: “Methodologically ... the ideal data for an application of AT consist of longitudinal ethnographic observation, interviews and discussion in real-life settings, supplemented by experiments”.

However, different issues present themselves for different levels of the activity structure.

AT models activities in a hierarchical way. At the bottom level are ‘operations’, routine behaviour that requires little conscious attention, e.g. rapid typing. At an intermediate level are ‘actions’, behaviour that is characterised by conscious planning, e.g. producing a glossary. There may be many different operations capable of fulfilling an action. The top level of description/analysis is that of the activity. Attempts to define this level are typically vague, reflecting the diffuse nature of the concept. Kuutti (1996, p.28) states: “An activity is the minimum meaningful context for understanding individual actions”.

For actions there is an emphasis on identification of simple goals, presumably ones that can be articulated. For operations the criterion is whether behaviours are habitual (automated) or not. While this is consistent with the general importance of consciousness for Sovietic Psychology, it is not an easy process to work the habitual/conscious distinction in practice. For activities identification is even more problematic. For example what identifies ‘homework’ as an activity (I.e. a basic unit) for school children or ‘carpentry’ for a builder? The answer would seem to rest on an analysis of cultural practices (deducing motives) and (implicitly) an appeal to everyday language categories (lending support to the identification of motives).

There have been a number of attempts to provide a structured approach to the application of Vygotsky ideas and their development within Activity Theory (although still heavily interpretative). There is no obviously unified approach and methods differ in their applicability to different levels (grain) of analysis. The most appropriate system of data collection and recording will also vary. Video-tape, interviews, observation of interactions, discourse analysis etc. have all been employed. The following illustrate the current diversity.

Bodker’s (e.g. 1996) approach to studying artefacts-in-use. This helps to identify levels by looking for (i) breakdowns - situations where unexpected problems occur; such as when the tool behaves in an unexpected way; this maps into the operation-action distinction and (ii) focus shifts - a change in the focus or object of actions or activity that is more deliberate than that caused by a breakdown. Bodker (1996) gives examples based on video-tape analysis.
Engestrom’s (e.g. 1993) analysis of ‘contradictions’. Engestrom argues that, from an AT perspective, there are three basic principles for recording and analysis of data: (I) taking a collective activity system as the unit of analysis (gives context to individual events); (ii) an historical analysis of the system (how it evolved to its current state); (iii) ‘inner contradictions’ of the system should be identified as the source of disruption, innovation, change and development (for example: between the patient’s changing problems and the rule that guides procedure for kinds of appointment).

Christiansen’s (1996) activity typology of tools. Based on the need to identify which artefacts are tools in which activity, and to identify a network of activities Christiansen maps the potential of the tool(s) for mediation into particular activities. In I-Mass we will use a version of these methods specifically arranged to support the design process of distributed system.

4.2 Functionality in the I-Mass system

This section focuses on the I-MASS functions to bridge the notions of activities and interactions between the user and the system.

4.2.1. Introduction

In the IMASS system the agents should bring librarian’s expertise to users. In the first place one should ask why to bring in such functionality in the system to the user. Normally an end-user is not very interested in the arcane knowledge of library stems, archival records or other instantiations of the cultural back office. More expert users, like librarians, archivists, curators actually do have this knowledge about their organizations. The knowledge that more expert users have about, for instance, the procedures and standards within a particular cultural heritage domain should be made visible in the system in order to help less expert end-users to find their way in the domain.

Considering the end-user, the need for the functionality of the build-in librarian to guide the user through the domain depends on the degree to which other functions have been realized in the user interface of the system. For instance, when the user interface allows only a small range of restricted choices to the user, the less there is a need for bringing the expertise of the librarian expertise to the end-user. If the user interface of the system offers a wide range of choices in a very flexible manner, end-user will have to know more about the system in the first place to exploit it fully. The same counts for the degree of knowledge offered about the knowledge landscape.

One can compare the choice for bringing librarian’s expertise to the end-user or not, with the choice that had to be made in the past between menu-driven interfaces and command line interfaces. In menu-driven interfaces, users do not have to recall the name of each an every command because it is only necessary to recognize the proper name of the intended command among the list of command names in the menu. In command-line interfaces users have to know both which commands are available and what names they have. As such, the menu interface requires less system knowledge among end-users in order to access the information domain, whereas command-line interfaces presuppose either a strongly developed system knowledge or an elaborate (on-line) help function or documentation.

From the hypothetical choice between the command-line interface and the menu interface, two implications follow concerning the I-Mass system. First, I-Mass' user interface should be designed such that it requires the least amount of knowledge from its users about the domain and about the system itself, particularly where it concerns I-Mass' non-expert users. Secondly, a minimal amount of required user knowledge
can be attained by incorporating such knowledge into the system. From this, it does not only follow that I-Mass should preferably offer a modern graphical or direct manipulation user interface, but also, stretching the implication to its limit, that a most usable I-Mass is one which incorporates and offers the expertise of the well-experienced librarian to its end-users.

4.2.2. I-Mass as a tool to access the cultural heritage domain

I-Mass is a tool which allows users to acquire access to the information and knowledge that is available in particular domains. Regarding the cultural heritage domain in general, it is possible to make the following distinction:

- the cultural heritage domain and its inherit organization
- access to the cultural heritage domain (metadata) and the ontology of the domain
- tools to access cultural heritage

It should be noted that although the distinction itself is evident, what the distinction actually refers to may be less clear and depend on the perspective and purpose that it is used for. A reference work, like the Oxford English Dictionary, for example, may be regarded as a cultural heritage object in itself, and, when used as a collection of definitions of terms, it may be regarded as actual access to the cultural heritage domain, and finally, since it needs being mastered before it may be used as a reference work, it is a tool. In similar vein, I-Mass is both a tool as well as a way to access (part of) the cultural heritage domain.

Making the exact distinctions between the domain, access and tools becomes more complex when attention is drawn to the constituents of the I-Mass system. It is clear that eventually, the domain of the I-Mass system is the cultural heritage domain as it exists outside the system and may be accessed by means of the system. It will also be clear that eventually, I-Mass' user interface is the tool to access this domain as well as to access whatever is relevant to access within the system, as far as the end-user is concerned.

In order to access information within a domain, end-users will have to know something about the domain and how it is organized, something about how to access the domain and the way in which accessing the domain works, and finally, something about how to use the tools and systems used to provide access. In order to find a particular book, it helps to know which books there are (in the domain), what the title of the book is (to access the domain), and how to specify the title of the book (to the search tool which provides access).

The whole point of making the distinction between the cultural heritage domain, accessing the domain, and using tools to access the domain is that, in principle, providing tools requires users to learn and master so-called secondary functionality. Searching and retrieving items in the cultural heritage domain is the primary functionality of I-Mass, but in order to do so, users are not only required to know something about the domain, but they also have to master the secondary functionality of how to use I-Mass itself.

In order to write letters, documents or whatever, first, it is necessary to learn how to use a pencil, a pen or a word processor. However, apart from the burden of having to master the tool, computer-based tools like I-Mass - and here the similarity between I-Mass and the pencil breaks down - may actually make it easier to attain the primary functionality by providing end-users with the knowledge and facilities about themselves and the application domain.
Different system (tools, systems, etc), like I-Mass, the local library or a real curator differ no only with respect to the size and nature of the knowledge domain that they provide access to, but they also differ in terms of how much knowledge or functionality they provide to the end-user to access these domains at the different levels of abstraction (cf. the domain, access to the domain, and use of the tools).

In the discussion of the menu system versus the command-line interface, it was already argued that the more knowledge, a search tool or user interfaces requires at one level of functionality, the less an end-user has to know about other levels. Comparing different systems or tools, well experienced real world librarian or curators would probably offer an easiest way to use and access the cultural heritage domain, since they know what there is in the domain, they know how to access it, and because they are intentional human beings themselves, the knowledge of how to use them is already part of the dialogue skills of the end users themselves. In order to determine which functions are necessary to implement the "well-experienced librarian" functionality to assist the user in accessing the cultural heritage domain, the remainder of this section will attempt to analyze what knowledge functions might be offered to the end-user by means of agent technology and which system functions are required within the I-Mass system to enable the envisioned end-user functionality.

### 4.2.3. Non-intelligent functions in I-Mass

The primary functionality of the I-Mass system is to enable users to search for items and references in the cultural-heritage domain. Within the realms of the non-intelligent systems, the usefulness and usability of I-Mass may be increased by providing secondary functionality to support the user's searching activities. From the literature on human searching and retrieval, three general findings are important here. First, the success in retrieving a particular item or set of items critically depends on knowing what else there is "out there". As such, it is important that I-Mass allows its users to acquire overviews of parts the domain. Secondly, even with full knowledge about the search domain, search and retrieval is more successful when people are allowed to gradually move towards the target rather then having to completely specify the search request at once. As such, apart from a down-to-earth query interface, I-Mass should enable its users to (re-)focus, restrict and expand searching to specific areas of the search domain as means to iterate towards the search targets. For the same reason, I-Mass should allow its users to temporarily store intermediate search results as landmarks and to reuse them on later occasions. Thirdly, people tend to evaluate the usability of interface higher when they allow users to adapt presentation aspects to their own preferences. As such, I-Mass should allow users to customize its user interface (adaptability), and to inspect, specify and adjust their interests, preferences and settings. Note that the effects of adaptability on performance scores are generally mixed since users often choose suboptimal settings, but, in principle, customization is a good thing.

### 4.2.4. Intelligent functions in I-Mass

In the previous section, the discussion was limited to the non-intelligent features of I-Mass and, more generally, to systems for searching and retrieval. The notion of adaptability was just mentioned, and in the context of more intelligent system, this notion may be further extrapolated, based on the premise that users often lack the knowledge and insights about the application or the application domain to optimally adjust any settings. Having the system take care of intelligently choosing settings.
allows the users to focus on their main tasks and it may help to avoid suboptimal settings.

First, I-Mass should enable that queries and the presentation of results are adapted to the implicit characteristics of the user, such as general expertise, expertise within a particular subject area, language and cultural background, and the use of particular interaction devices (adaptivity). In order to attain an adaptable interface, I-Mass requires a mechanism to monitor the dialogue with the user, a set of rules to infer the relevant characteristics of the user from the user query or dialogue, a user model or profile to store the findings, and an adaptation mechanism which adapts the interface to the user's characteristic.

Secondly, one step beyond regular or 'planned' adaptivity, is when the system starts to make inferences about the user's intentions and uses these to guide the query and present the query results (pro-activity). In order to implement pro-activity, it is not sufficient to merely observe the dialogue history guided by a set of rules. Rather, what is needed is a mechanism to make inferences about the user's intentions, based on a task or domain model which describes what a user may be doing to acquire a particular goal. In addition, what is required for plain adaptivity is a mechanism for drawing intentional inferences or hypotheses about the user's intentions from the user's activities as well as mechanisms to select the most applicable hypotheses, an adaptation mechanism, and a mechanism to evaluate the success and failure of the adaptation.

When intelligent features are incorporated into the system, there is no need to restrict their use to the system itself. Indeed, provided with a proper interface, the concept of transparency demands that such functionality is accessible to end-users, either allowing them to switch such facilities on and off, or to allow them to inspect any settings or to enable end-user programming and allow end-users to change or newly instruct intelligent agents by themselves.

One of the secondary user functions is to present an overview of the cultural heritage domain. These user functions are provided to the regular users as well as the knowledge providing users since they have to 'store' the items in the cultural heritage domain. Therefore we introduced the notion of the Knowledge Landscape, which in fact is also ‘build’ to provide the primary user- and system- functions. The Knowledge Landscape is a tool for the design and use of ontologies. The knowledge providing user has to be made aware of several terms for describing elements in browsing this Knowledge Landscape. The intelligent functionality of the Knowledge Landscape is the pro-activity characteristic of the agents and achieved by several mechanisms. The agent needs to have a fixed set of goals: they should provide facilities to draw one’s own knowledge landscape, they make users aware which reference work is used, automate tasks and learn (re-use) results from other agents to fulfill the user preferences.

Regarding all the I-MASS functionalities one agent-based requirement the system should have is predefined behaviours. These behaviours can be executed in parallel and adapted to the user behaviour. A problem that arises is how the agent should learn which action should taken next after the execution of a behaviour. Combining the selection mechanism for agent behaviours and Communicative Acts can be an important issue for I-MASS. Applying semiotics at the user interface to make the user aware how to use the agent behaviours and the Communicative Acts is a way to improve the user-friendliness. Examples: A user model agent can have a user option for automatic updates. In the behaviour of the agent is it possible to use the “subscribe” act. This act is used when we want another agent to notify another agent
whenever the object identified by the reference changes. Another possibility is to use the “query-if” act, which is used when an agent needs to know if a particular reference-work service is registered at the core system. Or for agents that are trying to find a set of agents to ask for help in certain classes of situations the “call-for-proposal” act could be used. The descriptions and how the FIPA communicative acts can be used found at (FIPA 00037, 2001) A kind of end user programming facility like in (deSouza, 1997) on the side of the knowledge landscape can be interesting since we want the user to draw it’s own knowledge landscape.

4.2.5. A retrospective of I-Mass functions

This section lists the different types of functionality discussed in the previous sections by way of summary.

Primary user function:
- to search and retrieve items in the cultural heritage domain

Secondary user functions:
- present overviews of (part of) the cultural heritage domain
- (re-)focus, restrict and expand searching to specific areas of the domain
- temporarily store intermediate search results as landmarks for reuse
- adapt presentation aspects to one's own likings and preferences (adaptability)
- inspect, change and specify preferences
- instruct agents for search and presentation purposes

Primary system functions:
- acquire the user's requests, interpret and execute them, gather and present results (primary functionality)
- monitor the user's activities
- interpret and adapt to any characteristics of the user, adaptable, adaptive and pro-active
- issue the search or query requests
- gather any results from the query
- adapt the presentation of the results
- present the user with the results of the request

Secondary system functions:
- create, maintain and update the user model; the user interaction record
- interpret the user's activities
- generate hypothesis about the user's intentions from the task or domain model
- test and select hypothesis about the user's intentions
- adapt the presentation to the interpretation results
- update the user model according to the most relevant hypothesis
- adapt the presentation of results according to the user model
- evaluate the results of any adaptations

Agent-based requirement:
• the agents need to have pre-defined behaviours to provide query/response services

4.3. Sketching I-MASS Human-MAS interaction (scenario-based interaction models)

In the following some examples of results are reported, to express the kind of outcomes that can be expected from activity analysis that was conducted within the Content Providers’ domain-experts users. In particular, relevant aspects of the selected activities will be reported, as well as tools used to carry out the activity itself and places where it can potentially take place. Moreover, a macro-scenario is reported as preliminary data on which the following phase of interviews in the context of work will be focused (WP2).

4.3.1. Research activity: an overview of relevant Aspects, Tools and Places

Research activities are potentially long-lasting processes which can involve a variable number of subjects and can be supported by various kinds of tools. They often take place in several different places, according to the kind of information available on each site, the nature of the information sources and the available services.

In the following some considerations about the nature of the activity investigated are proposed, focusing on the distributed aspects, the influence of personal expertise and the role of pre-existing knowledge, strategies and best practices.

4.3.2. Research as a distributed activity

Research activities can be considered as distributed along different dimensions.

In order to perform a research activity on a given topic, users have to accomplish different kinds of sub-activities. Just to list some of them, researchers need to identify the possible space/context of their current research, to implicitly formulate an (some) information need(s) and look for the available information sources. Once potentially interesting sources (e.g. a book) have been discovered, a micro-research is then performed on the specific information source (e.g. a book, an article, an archive…), in order, for example, to define the relevance of such a source for the higher level research activity, before actually gathering it.

Research is distributed in space. People search and retrieve information from different “data sources”. In many cases these need to be physically reached (e.g. a specific library, a museum, …). In addition, in many cases, even knowing whether an institution owns and provides access to a given resource, direct contacts with the maintainers or front-office services (e.g. libraries without on-line catalogues, private collections, local museums…) are often required.

When the source has been accessed it needs to be evaluated. Such sub-activity is often performed in non-work contexts, like home, train, library, bookstore.

Research is distributed in time. We observed that the time span devoted to the research on a given topic depends on several factors. Among them, the objectives of the research, the accessibility of data, the time constraints, the complexity of the research domain, the availability of previous researches conducted in the same domain or about close topics. The way in which such factors vary and interact with

---

4 These results are the output of Unisiena field research at CRIBECU in Pisa with domain-expert users, namely research professionals.
each others, affect the time span needed to accomplish the research. The time span can vary from few hours to weeks, months and even years.

Research can be socially distributed. Often research activities are conducted in collaborative environment, where different actors (with potentially diverse backgrounds) work in teams. The social nature of the research activity can take different forms, from well structured tutoring-based collaborations to peer-to-peer interactions to accomplish a common/shared goal. Even in this case, the personal goals, practices and expertise play a crucial role in the co-operation.

Research activities are supported by very different kinds of tools, each covering specific spaces of the problem. Although such mediation tools are usually adopted in parallel, there is currently low integration among them mainly because of the different nature of the storage media (some of them digital, others paper-based, some others hardly accessible due to their physical characteristics, e.g. manuscripts, paintings, sculptures, copyright-protected texts, …). In addition, the availability of tools in specific research contexts (office, library, museum, …) strongly affects the ability of researchers to have at their disposal what they need in each context (from copy-machine, to fax, to digital texts and catalogues to easily query).

Research activity can spread over different knowledge domains (from art to history, to cinema, to anatomy). On the basis of the nature of the topic under investigation, in most cases, the actors face unfamiliar domains that require them time and additional research to get acquainted to new concepts and features. Indeed, cross-fertilization is very usual in many specific research domains.

4.3.2.1. Expertise

Researcher’s background knowledge significantly affects the ability to access and manipulate the information needed. People’s background knowledge on the research domain influences the way information is searched, gathered and manipulated, the ability to navigate through the proper information sources, the way concepts are perceived, interpreted, elaborated and combined into new knowledge.

What expressed above has a particularly significant impact on researcher’s ability to move across the different levels of knowledge representations, both within the same knowledge domain and across more or less interrelated domains.

At a different level, a crucial role is played by user’s skills in using the media which support the research activity, both typical of a given knowledge domain (e.g. iconographic inventories, specific thesauri, …) or mutually available in all the potential research domains (on-line catalogues, web-based services, manuscripts, microfilms…).

4.3.2.2. Personal strategies and best practices

Personal attitudes can be considered the way in which personal strategies in information seeking and knowledge management activities are performed. In general, personal strategies and best working practices affect not only the way information seeking is leaded, but also the modalities with which people manage, interpret and organize the information gathered in order to build new knowledge. Just to provide an overview of some relevant points individualised during the University of Siena field analysis, here follow some examples of best practices reported by the subjects and further elaborated in order to identify their rationales.

Most of the subjects interviewed have pointed out the role of information organization in their personal libraries and archives. Every researcher has personal rules and strategies for collecting data, information sources and references. Through storing
activity people indeed establish relationships among all the pieces of information collected, creating meaningful links that define the personal space of the problem. What clearly comes out from research activity is the meaningfulness of the folding strategies in following the main object of research activity as it is performed by the subjects.

Research activity can be described as a step by step process. The role played by intermediate results of sub-activities is complex. On the one hand, intermediate findings are actually perceived and used as accomplished micro-research outcomes (like performing an experiment, writing an article or review on a specific topic...), which are fundamental or accessory parts in the economy of a broader research goal (e.g. writing a book, a PhD dissertation, a cycle of lectures, writing a dictionary). In other cases partial findings are just meaningful in the context of the whole research, and need to be linked to other partial findings and to be placed within the scheme of the whole research activity. From this perspective, they can be considered the starting point for restructuring the space of the problem, the whole research through unexplored paths.

The context in which the research is performed significantly affects the organization and re-use of temporary works and findings, as well as the strategies for accessing intermediate results are part of the personal background of the researchers.

Some researchers have the inclination first to define the space of the research and the goals to be achieved right at the beginning of a research activity. Others use to collect broad sets of material first, analyse them and then address the research according to what they judged relevant, interesting, new according to their main objective, individualizing step by step new precise goals and modalities of work organization.

An operational research, e.g. aimed to produce some working document, can start from a general to-do-list which covers almost all the steps the researcher estimates as necessary to accomplish the goal of the activity. Starting from the firstly defined list new relationships among concepts and preliminary findings can be made in a process of continuous re-design / assessment of all sub-activities that have to be performed to accomplish the overall goal.

### 4.3.3. Scenario-based design

Scenario-based design (Carroll, 1995) is a method that allows designers and users to describe existing activities or new activities that can be produced by the interaction with a new artefact to be envisioned. A scenario is intended to detail a usage situation and to document step-by-step actions performed by users (Lewis & Rieman, 1993). It is usually represented by textual, pictorial or diagrammatic descriptions and can be applied in different stages of the development process of an artefact. Nielsen (Nielsen, 1995) states that scenarios can be used throughout all the system development life cycle and discusses seven applications including:

- diary scenarios for data gathering,
- brainstorming to envision new features of the system,
- scenarios for design and prototyping,
- scenarios for heuristic evaluation,
- stereotypes scenarios for data analysis in exploratory studies,
- scenarios for task-based user testing.

In I-MASS we decided to use a scenario building technique to clarify our thoughts about what the final prototype should be and its potential features and capabilities. In
particular we used scenarios for requirements gathering and prototyping of early ideas.
The analysis reported above inspired the definition of macro-scenarios which represent potential macro-activities as targets of the design of the support system. In the following we report an example of macro-scenario.

4.3.4. Macroscenario: the “point of view” in Virgilio’s work.

General description
Maria is working at her PhD dissertation.

Actors: Maria and her Professor.

Situation and environment
Maria is a student who works most of the time at home or in the library of her faculty.

Task context
The contents to be added to the thesis have to be chosen according to several concurrent criteria:
Maria wants to write an original dissertation
Maria would like to write a dissertation on Virgilio
The title of the dissertation is “The point of view in Virgilio’s literature”.

Exceptional circumstances:
Finding and original reading on Virgilio’s literature.

Cinema represents a source of inspiration for her work.

The scenario
Maria is a Ph.D. student involved in Latin literature’s studies. She is working at her thesis focusing on Virgilio’s works and she would like to find an original reading on this topic.

It is Saturday night and she decides to go to the cinema to watch the last Vinterberg’s movie, “Festen”, without knowing anything about this director and his shooting style.

While she’s enjoying the movie, she notices the constant changing of points of views according to each different characters.

This leads Maria to recognize that this particular device is also suitable for Virgilio’s narrative style, because his characters, even when not central in the plot, have their own view of the story occurring.

The day after she goes to her tutor to discuss her new idea. Her tutor is enthusiastic about Maria’s idea and helps her defining her research theme as “The point of view in Virgilio’s literature”.

Afterwards Maria starts to look for sources that can provide her with the sufficient background on cinematography and the concept of point of view. This activity leads Maria to browse a new domain, she begins watching films by the same author paying attention to the use of the point of view, she deepens her knowledge about movie techniques and reads specialised books and journals on the topic under investigation.

She also meets cinema studies experts, among her friends and the University staff.

This activity is not strictly related to Virgilio’s literature, but it’s useful for Maria to collect the necessary knowledge on which to build her own interpretative tools to approach an original dissertation.

At the end of a long and tiring activity Maria will write an original dissertation about what a lot of researchers have previously studied and written applying classical paradigms.

In the following figure the activity above described is depicted giving an overview of the entire process divided in significant sub-activities and their links through time.
Writing an original PhD dissertation on Virgilio's Eneide

**Discussion**

- Defining the subject
- Defining the Hypothesis

**Looking for an original point of view**

- Searching and finding references on Latin literature
- Searching and finding references on Virgilio's work

**Getting an original hypothesis**

- Reading and using the selected literature

**Watching "Festen" movie**

- Saturday evening May 18th 2001

**Making new connections**

- Speaking with the tutor about the new idea
- Searching for cinematographic literature

**Temporal dimension**

2001

Reading text of French cinematographic criticism

2002

Writing an original dissertation

Model of activity. The "point of view" in Virgilio's work.
The table reported below shows how to map activity analysis with the individualisation of some functional requirements the system should implement in respect of the idea to support users’ activities.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Functional requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maria is working on Virgilio’s literature to prepare her dissertation.</td>
<td>Tools to access domain by subject title/author/references/temporal dimension</td>
</tr>
<tr>
<td>Maria is looking for an original point of view</td>
<td>Need to show un-expected relations among pieces of information belonging to different domain (Latin literature and cinema)</td>
</tr>
<tr>
<td></td>
<td>Need for tools that allow/suggest cross-cultural researches</td>
</tr>
<tr>
<td></td>
<td>Needs for tools that are able to show the meaning of cross-links proposed.</td>
</tr>
<tr>
<td>Maria is an expert of Virgilio’s literature and a novice of cinema techniques</td>
<td>System has to be able to support in different way Maria according to the different level of expertise of the domain she is investigating.</td>
</tr>
<tr>
<td></td>
<td>Need for suggestion to individualise best experts in the field</td>
</tr>
<tr>
<td></td>
<td>Needs for discovering if similar research has been conducted by other researcher on the same or on related domains (i.e. point of view in painting and movie direction)</td>
</tr>
<tr>
<td>Maria need using sources that distributed in space and time</td>
<td>Need to provide content interoperability to guarantee remote accessibility</td>
</tr>
<tr>
<td></td>
<td>Need to access different content through the use of the same tool (i.e. director for films and author for paper, writing techniques for literature and direction techniques for cinema).</td>
</tr>
<tr>
<td></td>
<td>Need to access primary and secondary references (i.e. accessing books about direction techniques as well as accessing related movies that show the application of some particular techniques. It is the same as reading critical literature on a painting while seeing the painting itself).</td>
</tr>
<tr>
<td></td>
<td>Need to support different format (i.e., papers, images, movies, audio, drawings, maps...)</td>
</tr>
</tbody>
</table>

In this chapter we have described the methodology applied to conduct our field activities and the first results collected. Those results have been presented in form of macro-scenarios of the observed activities.

The scenarios described above together with the coming results from further activities analysis will be the basis on which to design a series of maps of activities. More
precisely each of the maps will be representative of the different users’ profiles already selected. What we have just described will allow us:

- to individualise a general model of the activity
- to define together with all of the partners of the consortium the most appropriate phases of the macro-scenarios that will become micro-scenarios
- to individualise the requirements of I-Mass system on the basis of the selected micro-scenarios.
5. CONCLUSIONS

The most important concepts in this report are probably (macro- & micro-) scenario, interaction model, dialogue structure (interaction protocol), ontology, mediated exchange of structured knowledge, communicative act, shared meanings, learning interface agent, self-describing service agents, co-operating heterogeneous knowledge sources, explanatory discourse, and adaptivity. In the following we will provide a perspective from which these concepts' importance can be shown for the I-MASS system design task, particularly for I-MASS prototype II. This perspective will be given also in the form of a scenario, a meta- or design scenario for building I-MASS prototype II “User Interaction”.

First, the whole idea of having an explicit Interaction Model (a model situated between user model and domain or core system model) is to mould the enormously rich field of possible interactions between (kinds of ) users and the prospective I-MASS Core System in a model, which partitions the field of interactions in as much declarative components as appears to be feasible. Both the partitioning and the declarative building blocks for ways (and modes) of HC-interaction constitute in itself chunks of knowledge that may be assigned to particular kinds of agents, taking care for the appropriate use of that knowledge in the course of interacting. Based on the knowledge that these agents possess they are also assigned a behavioural spectrum from which they may activate particular behaviours as intended responses to the performatives received from other agents. In addition, these agents also possess a goal structure that allows them to (semi) autonomously stay involved in trying to satisfy local goals (that the I-MASS designers view as important from a more global design standpoint).

The Interaction Model should be devised according to what is known about the User Model and the Core System Model. The first model can partly be based on existing knowledge about the kinds of users that may want to use I-MAS and partly on specially elaborated Macro-Scenario's that may be seen as generic scenario's for a class of users. The second model is partly based on the already built system for accessing local Information Sources (provided by content providers) and the prospected functionalities that have to be added, like Knowledge Landscape, Virtual Reference Room, Syntactic Interoperability (Meta-Data Registry) and Semantic Interoperability (ensuring contextualisation and meaning exchange between different Information Sources). Here the Knowledge Landscape functions as the I-MASS internal structured repository of cultural objects and their interrelationships, of rules enforcing specific forms of cultural knowledge organisation (knowledge organisation rules), and of structures that are descriptive of the knowledge landscape itself and the Core System built around it (introspective knowledge). An important aspect of the Knowledge Landscape is its Ontology, i.e. the store of conceptual descriptions that are basic to any more complex piece of cultural knowledge represented in the Knowledge Landscape. This KL-ontology is normative in the sense that any piece of knowledge (partial questions and partial answers) on their way to and from the local Information sources receives its meaning purely from its embedding in the KL (and its underlying KL-ontology). Which mechanisms are responsible for that embedding does not matter here (we assume, of course, that those mechanisms make sense against the background of VRR, SynI and SemI).

Ideally, any User uses only a part of the KL-ontology to formulate her/his questions and to receive her/his answers. However, even when this is not strictly the case it is
useful to introduce the notion of User-Ontology that functions as the conceptual framework within which the User makes and receives her/his formulations (of questions and of answers). Then the Interaction Model is responsible for a mapping (translation) between User-Ontology and KL-ontology (and in reverse) in such a way that the formulations in the KL are normative (for what the User really asked and for what the User really receives as an answer!). Since it is clear that this system approach may introduce disparities and/or unintended biases between what the User meant and what the system appears to understand, there are a number of feedback mechanisms required at the User-side in order to bridge any semantic gaps. Moreover, there may be cases in which the User should even be allowed to consult the KL directly in order to experience what "happened" in the processing in-between User and Core System.

This will become the key issue in the design of the I-MASS interface since we will need new interaction modalities so to allow the user to work on metadata (e.g., the KL-ontology) in a way that would allow the user to develop knowledge and expertise on this new domain of activity and finally to shape this domain through his work. Only in this way we could build again the illusion of user control so to allow him to trust and use confidently the system.

At this stage of the I-MASS design it is, however, more important that the User should not be forced to formulate too much in order to get sensible answers from the Core System. That means that in the Interaction Model particular forms of interaction support are built in that assist the User in being as succinct as possible in his formulations. This interaction support may take at least three forms:

a) automated inferences of what the User intends;
b) automated assignments of interaction parameters;
c) automated evaluation of the running interactions, and their adaptations.

All of these forms may be covered by specific kinds of agents that have been given particular chunks of interaction knowledge, behavioural spectra, and individual goals to reach for in the absence of any particular consultations by other agents. In this way the Interaction Model is populated by at least Interaction Inference Agent, Interaction Parameter Agent, and Interaction Evaluation Agent.

The field of all possible HC-interactions may be partitioned in generic dialogue structures covering high-level ways of approaching the I-MASS system and expected renderings of this system. While doing that it will become clear that an orthogonal perspective of cultural - HCI-signs, organisation layer, and environmental layer - may prove useful in finding repetitive structures involving different kinds of cultural knowledge. Precisely the elaboration of Macro-Scenario's into Micro-Scenario's will allow us to explore and identify the kinds of cultural knowledge involved. Parts of this analysis have an impact on the User Model, and other parts have a bearing on the Interaction Model. Furthermore, the layer of HCI-signs should guide the design of the User Interface for I-MASS, in particular it should allow to specify possible way in which the interaction modalities could extend beyond the command line interface and the GUI interface.

In fact, the Scenario may be viewed as an initial spec of the required interaction ways and modes, and hence as constitutive for the Interaction Model.
From the outset it is useful to separate the User and her/his Query, and correspondingly the User Model and the Query Model that is built up directly after the User's initial formulation (by means of the User Interface). Accordingly, a User Agent and Query Agent are erected. The processing of the Query on its way to the KL is dominated by contextualisation procedures that ensure a final embedding of the Query in the KL. From there the assessment of which Information Sources to approach is taken, mostly without User control. The Query Agent also allows to present to the User an elaborated, contextualised representation of the initial Query that is in line with the final, internal representation in the KL. When the User is not satisfied with the new Query formulation or needs more explanations, then an 'explanatory discourse' is set up in which the User may provide feedback that adapts the KL-representation of her/his Query. It is clear that the self-descriptive abilities of the KL are important here.
6. REFERENCES

Chapter 1


Chapter 2


Castelfranchi, C., Engineering Social Order, (National research council – Institute of psychology)


Edmond, B., “Adaptive User Interfaces Based On Activation Levels For Knowledge Exploration And Construction”, National research council Canada


Horwitz, E., Agents With Beliefs: Reflections On Bayesian Methods For User Modelling, In Sixth International Conference On User Modelling, Sardinia, 1997.


Leontjev, A.N. (1923) Journal of Genetic Psychology, 40, 52-83


Chapter 3


Bylund (1999) Markus Bylund, Coordinating adaptations in Open Service Architectures


Dieterich et al. (1994) Hartmut Dieterich, Uwe Malinowski, Thomas Kühme, Matthias Schneider-Hufschmidt:State of the art in adaptive user interfaces


Dieterich et al. (1994) Dieterich, H., Malinowski, U., Kühme, T., Schneider-Hufschmidt, M., State of the art in adaptive user interfaces


JADE, JAVA Agent Development Framework, [http://jade.csel.it/]


Lashkari et al. (1994) Collaborative Interface agents, Yezdi Lashkari, Max Metral, Pattie Maes, Proceedings of the Twelfth National Conference on Artificial Intelligence

Labrou et al. (1999), Yannis Labrou, Tim Finin and Yun Peng, The current landscape of Agent Communication Languages, [http://citeseer.nj.nec.com/labrou99current.html]


Chapter 4


7. SUGGESTED BIBLIOGRAPHY

Cognition
A. Lee, Social navigation: design for environments to Foster electronic marketplace communities, (IBM TJ Watson research center)
C. Castelfranchi, Engineering social order, (National research council – Institute of psychology)
M. S. Ackerman, Metaphors along the information highway, in “Proceedings of the symposium on directions and impacts of advanced computing”, Cambridge, MA, 1994
M. S. Ackerman, R. T. Fielding, Collection maintenance in the digital library, in “Proceedings of digital libraries ’95”, Austin, TX, June 1995, pp. 39-48
S. Grant, Modelling complex cognition: contextual modularity and transitions, ..., ...
http://www.simongrant.org/pubs/um94/text.html
E. Hutchins, Cognition in the wild, Cambridge, MA, MIT Press, 2000
http://tonic.asc.unl.edu/omlit/Hutchins_1995.htm

User modelling - web interfaces
Ackerman, M. S. Providing social interaction in the digital library, in “Proceedings of digital libraries ’94”, College Station, 1994, pp. 198-200
Chi, Ed. H., Pirolli, P., Chen, K., Pitkow, J., The scent of a site: a system for analyzing and predicting information scent, usage and usability of a web site, (Xerox Palo Alto research center, CA)
Chi, Ed. H., Pirolli, P., Chen, K., Pitkow, J., Using information scent to model user information needs and actions on the web, (Xerox Palo Alto research center, CA).
Edmond, B., Adaptive user interfaces based on activation levels for knowledge exploration and construction, (National research council Canada) http://cbl.leeds.ac.uk/~jas/emond.html
http://chat.carleton.ca/~schand2/agents/ui.html
http://www.contrib.andrew.cmu.edu/~plb/wwwum99_workshop/stefani.html
http://www.contrib.andrew.cmu.edu/~plb/wwwum99_workshop/wilkinson.html
Agent interfaces: why we need them, ..., ...
Kaaasinen, E., Usability issues in agent applications: what should the designer be aware of, USINACTS 1.6.1998, Finland
http://atwww.hhi.de/draft/usinactsnorm/project/agent.htm
Keeble, R. J., Macredie, R. D., Assistant agents for the world wide web - intelligent interface design challenger, in “Interacting with computers” n. 12, 2000, pp. 357-381
Lewis, M., Designing for human-agent interaction, (School for information sciences, University of Pittsburgh)
Lieberman, H., Shearin, S., Intelligent profiling by example, (MIT Media lab)
www.elsevier.com/locate/knosys
Twidale, N., Nichols, D., Designing interfaces to support collaboration in information retrieval, in “Interacting with computers” n. 10, 1998, pp. 177-193
www.elsevier.com/locate/intcom

Agents
Ackerman, M. S., Starr, B., Pazzani, M., The Do-I-care agent: effective social discovery and filtering on the web, (Information and computer science, University of California)
Conati, C., Klawe, M., Socially intelligent agents to improve the effectiveness of educational games, (Dept. of Computer science, University of British Columbia, Canada)
Ferber, J., Multi-Agent System: an introduction to distributed artificial intelligence, Addison Wesley Longman, Harlow, 1999
Maes, P., Agents that reduce work and information overload, (MIT Media lab, Cambridge, MA)
http://www.cs.unimaas.nl/p.vogt/csr.ps.gz