The role of interface mock ups in establishing common ground in a distributed development team

A. Woodcock  VIDE Research Centre, Coventry School of Art and Design, Coventry University, UK
S.A.R. Scrivener  VIDE Research Centre, Coventry School of Art and Design, Coventry University, UK
S. Clark  VIDE Research Centre, Coventry School of Art and Design, Coventry University, UK
X-H. Sun  VIDE Research Centre, Coventry School of Art and Design, Coventry University, UK

Abstract
This paper uses a software development project, comprising of end user representatives, evaluators and developers to consider the value of an interface mock up in establishing common ground within a large, distributed software development team. It introduces the scope of the project - to develop an integrated, web-based, platform to support team working (UNITE) - and then considers the role of the mock up of the interface in helping the team understand the underlying concepts behind the project, and in facilitating discussion, agreement and understanding between stakeholders. Lastly the paper evaluates the success of the mock up in achieving these aims.
The role of interface mock ups in establishing common ground in a distributed development team

Introduction
The UNITE project has been funded by the EU-IST Programme for 2 years and concerns the development of a web based, Ubiquitous and Integrated Teamwork Environment to support the needs of mobile workers, acting as members of virtual project teams. This requires the integration of traditional computer supported co-operative working (CSCW) tools (e.g. joint calendars, video conferencing, email) in a seamless environment structured around tasks and communication needs. The development of such an environment involves the contribution of a number different research groups from UK (Coventry University), Holland (Pentascope), France (IBM and Steria), Germany (FhG-IAO and FhG-SIT), Portugal (ADETTI) and Israel (IBM) with different groups responsible for software development and integration, evaluation and user testing.

Since there was no pre-existing system, a Semantic Model (SM) was developed to forward an underlying understanding of the proposed system, and to allow communication between the different stakeholders (especially developers) at the early stages of the project (i.e. before and during the iterative development of the system) about the UNITE concept and its implications. The SM was used to drive, and became embedded in, the interface mock up (a version of which is shown in Figure 1). The role of the SM was to provide a framework to guide development and to provide a communication environment and overview to facilitate the rapid development of a coherent system.

The SM defines a set of concepts and a basic framework about what the UNITE platform should be (see below for more details). The inspection of the SM and interaction with the mock up (in particular) could potentially enable end user representatives to talk to the developers (in a shared language) about the functionality and where the mock up did not meet their previously described requirements.

Previous papers (e.g. Reinema. et al 1998 and 2000, Woodcock et al, 2002) have described the organization of the consortium, the UNITE concept and the progress of the project towards developing an integrated platform. In summary, UNITE provides a platform which integrates different collaboration modes and requirements, and provides a project/task centred environment allowing its users to concentrate on the task in hand, rather than becoming distracted for example, by having to memorise the addresses of team members, their location and preferred working practices, and application centred information. Additionally, as a web based environment, the platform is available for users wherever they happen to be working. By providing a virtual working space for all project members tacit knowledge (Polanyi, 1966) can also be spread more easily around the group – so promoting group learning, awareness and camaraderie between a team which may rarely meet face to face (Woodcock et al, 2001).

This paper considers the role of the mock up during system development with regard to its embodiment of the SM and as a communication support for encouraging interaction and reflection amongst the project members, i.e. as a means of establishing common ground between project members. The paper is divided in to three parts. It commences with a section outlining the interface and its key components. From this, we will introduce the SM, which is the theoretical basis of the system development. The rest of the paper examines the nature of discussions centred on the interface and will present the results of a survey of team members to consider whether the interface was successful in achieving its aims.
The interface mock up

The need for a mock up interface was postulated in the first quarter of the project. The end user representatives wanted a physical representation of the UNITE concept as they found the written specifications of the requirements and the SM hard to understand and relate to practical experience. The developments of the mock up culminated in the production of a web-based demonstrator system, see Figure 1 below, which developed into a vehicle for discussing the Semantic Model and its implications for user behaviour.

As can be seen from Figure 1, the system offers a message centre, document repository, shared calendar, address book, bookmarks, meeting management, support for communication and collaboration, and security services centred around the needs of the project team. The message centre for instance presents a unified view of all forms of messages, whether these are voice mail, email etc enabling all asynchronous communication between team members to be sent and received from one place. For synchronous (real time) communication, users gather in 'collaboration areas' which are defined by the project team at project initiation, to suit their needs and the communication preferences of the team members. These are shown as discrete areas (top right) of the interface where different collaboration 'scenarios' are available, without showing the technical details of the required tools and services. By entering one of these collaboration areas, services such as text chat and application sharing are automatically opened and become available to everyone who has been invited to and enters the area.

Figure 1: The UNITE interface
A UNITE virtual workplace as shown by the mock up provides users with a number of tools or services, which can be assembled from a pre-existing reservoir of current services. Each project manager, when initiating a new project, can select from these services those facilities that best suit the needs of the project and its members. For example, a project team may consist of people, who spend a lot of time travelling, and working from different offices, using different communication tools. In setting up the Virtual Project Office (VPO), the contact details and preferences for each team member are added to the system and the services and tools required to contact individuals will be configured. This means that if you go into the VPO, for example to discuss a forthcoming meeting with colleagues, the UNITE platform will contact them for you whether this is by phone, email, SMS, or text chat. You are thereby freed from the onus of trying to remember different phone numbers, the location of the rest of the team, etc., and so can remain focussed on the task in hand. Also, by using one of the collaboration areas, a predetermined set of tools and services is provided for use by your team (for example, application sharing, videoconferencing) without any of the team having to select and open applications.

The platform hides the details of each tool or service behind a unified user interface, manages them and provides a consistent look and feel across all elements, functional modules, and windows. It clearly indicates a user's location in the Virtual Project Office (VPO) at all times; in particular, it indicates the user's current collaborative work context unambiguously. The VPO offers users awareness and intuitive navigation like a real team office, but without imaging an office in 2D or 3D. The majority of user activities take place within the Web browser window, although collaboration tools do launch their own application windows. At the moment, UNITE users can access the platform from a desktop PC via the UNITE project Portal using a regular Web Browser. In the future, they will also be able to use mobile phones (e.g. WAP browser) or other portable devices (e.g. PDA).

As each development partner (i.e. IBM, Steria, FHg-SIT, ADETTI and Coventry University) is responsible for the development of one or more modules (Figure 2), the integration of the elements into a unified and consistent whole was paramount. In order to reduce the inconsistencies and maintain an overall framework for the project a semantic model was used to guide development.

![Diagram of the basic platform](image-url)
The Semantic Model

It is not the aim of this paper to discuss the semantic model, or its adequacy as a representation of teamworking. Rather we are concerned with its usefulness in helping the project team understand the basic concepts behind UNITE and their implications.

The semantic model (shown diagrammatically) in Figure 3 was used from the start of the project to specify a set of relationships which governed the behaviour and functionality of the UNITE platform. It is its theoretical base and deals with context closure in team collaboration. It defines two kinds of work contexts and their relationship: the personal and teamwork context. For the users, the personal work context contains a set of tools and interfaces just valid for the individual; the teamwork context contains collaboration services and interfaces built from another set of tools and shared with the team members within one project. For example, when in the personal context area (the circles in Figure 3), the user is able to get into contact with their Personal Assistant (for e.g. messages from a teamwork context), export resources to a teamwork context, and select to enter a teamwork context (also called Virtual Project Office). Likewise the SM gives rise to the following types of relationships:

- when at or from his/ her personal desk in the Virtual Project Office the user can see all team members, get contacted by others, access all project resources, change to a collaboration area or move to another personal desk to initiate communication.
- when meeting at a personal desk, each member has to agree to the collaboration and access to each others resources.
- when meeting in a collaboration area: the possibilities are defined by a collaboration scenario and available collaborative resources. These are under the control of the team.

Clearly, if the development team were to produce an integrated, coherent and usable system they all needed to understand the Semantic Model and its implications. The SM also had to fulfill the requirements of the end users and help in their instantiation in a usable and effective system. Although it was not essential for the user representatives and evaluators to understand the SM it was believed that its accurate embodiment in the user interface would enable a coherent, integrated
system to be produced which would facilitate the acquisition of a user model of the system and reduce the number of usability issues. The mock up was one of a number of vehicles for doing this.

**The role of the mock up in the project**

To gain insight into the role of the mock up, a questionnaire was sent to the technical and management leaders of the project to determine what role they thought the mock up had in the project and the extent to which they believed it had fulfilled that role. The mock up was believed to be able to fulfill the following roles:

- to guide the project team in sharing a common understanding of the UNITE platform. For example, ‘experience with previous projects had shown that paper descriptions were always insufficient to visualise implementations ahead of time, thereby leaving people with their own interpretation. This may be due to the fact that people do not take the time to sufficiently analyse paper documents, and the difficulty of representing something that does not already exist. Looking at a realisation of concepts is much more efficient.’ G.Lacoste, project manager
- to save resources. Without mock ups, assembling prototype implementations would be more costly/ elaborate and difficult because of lack of common understanding. While the actual implementation has to realize a specific set of features and services down to the fine details and may be restricted in scope by limited time and resources, the mock-up could present a gross overview and be more free in sketching features and services.
- to enable the refinement of the user interface through iterative development and feedback.
- to be the forerunner of an actual implementation and thus guide developers in implementation planning. In an implementation such as UNITE’s (which is research driven), a lot of decisions have to be taken in the process from design, specification, coding, unit tests to final integration tests, which cannot always be predetermined. A mock-up can help in such situations to clearly recall what the original design specification was - at least from the user interface point of view; it can help in the experimentation with alternatives and their consequences; and it allows feedback on these.
- to be a promotional tool. The mock up should allow light-weight demonstrations of the UNITE platform without the need of the surrounding, heavy infrastructure (Internet access, servers of many types, etc) thereby demonstrating to potential users what a UNITE implementation could provide to them in terms of benefits, functions and services and help users in preparing for an installation.
- to help users understand advanced features. It could help in discussing with users which specific features, out of a set of possible ones, would have which effect, and would be appropriate for them. It can help to explain to users that a current implementation is just an intermediate step in the evolution of more powerful versions.
- to serve as a bridge between the semantic model (SM) and the actual implementation. The SM provides a formal framework for the purpose of specifying and verifying the basic concepts of UNITE platforms, the mock-up although it draws on the SM, is more informal and is put in terms that are closer to users.

To summarise the mock up was supposed to facilitate understanding within the project team, act as a promotional tool, provide a means of interpreting the SM and provide insight into future systems.

**Assessing the role of the mock up in the project**

The evaluation of the utility of the mock up by the managers and the end user representatives showed that it was not thought to be as useful as it could have. This might have been due to the difficulties in its development and relatively late manifestation in the project lifecycle.
Whilst everyone had agreed that the mock up would be a valuable tool in the project, its actual role was controversial, and indeed some of the intended functions became contradictory. For example, the underlying semantic model was not well enough understood by the developers. This lack of understanding became manifest during the production of the mock up, derived from information in the project documents. The mismatch between the mock up and the SM meant that the mock up had to be more closely aligned to the SM before it could be released to the user representatives. This alignment was brought about by pre-releasing mock-ups to developers, who would engage in their own iterative development cycle, prior to release to the user representatives. This lengthened the mock up development time, and deferred in depth user representative involvement in project development.

The third version of the mock up (shown in Figure 1) allowed the basic concepts of UNITE (personal context, teamwork context, personal desks within teamwork context) to become clear to all developers and from then on it was used as a common base. The iterative development and achievement of a common understanding took longer to establish than had been anticipated, so there was not enough time left in the given timeframe to get feedback from the users on either the GUI or the Basic Platform functional view. In terms of easy trials the mock up partly achieved this, as it was possible to show users efficiently what UNITE can be.

Communication regarding the mock up interface
In terms of communication and management, the two-year project has scheduled quarterly face-to-face meetings, supplemented by more focused workshops to address user/evaluator or developer issues, and teleconferencing sessions on a needs basis, on-line chat and uses BSCW for document sharing and email. In this section, we examine some of the emails to determine what issues were discussed in relation to the mock up and analyse the transcript of a teleconferencing session to illustrate the way in which the use of the mock up promoted further discussion amongst the team.

During the development of the mock-up, one set of emails centred on the discussion of the meaning and understanding of the Semantic Model. Of these, a sample of 155 emails was collected during the first year of the project. In terms of the project lifecycle, the emails occurred at the stage when the mock up was conceived to the time when the first version was mature enough to be subjected to initial evaluation, and a partial understanding of the semantic model had been achieved within the project.

The emails were analyzed to determine who initiated discussion, who was involved in the discussion, what was its focus and severity, the cause of the discussion and whether issues were resolved. 40 of the emails focused on the semantic model and the mock up. In all instances the developers initiated discussion, e.g. by proposing a mock up to the agreement of the project (especially the end user representatives). In the iterative development stages, when developers produced a new version of mock-up, the users commented on it, sometimes resulting in conflicts. Such comments related to interface features, such as the size of text, online help, colour of background, and location of some functions; project management and the semantic model. Also specific issues were raised indicating that the mock up was not seen as a true representation of the SM such as “the mock-up does not separate personal work context and team work context clearly from each other”, and “Ambiguity in collaboration areas” was noted. Important, deep issues were not resolved through emails but at face to face meetings, or through teleconferences.

Crucially, the analysis of the emails (and the questionnaires) regarding the interface shows that the semantic model was not well understood at that time, and that a common ground from which to proceed and resolve misunderstandings could not be successfully established asynchronously. This
necessitated additional teleconferences and workshops, some of which were minuted and published as open documents on 'BSCW' - the shared project's document repository.

Discussion showed that a significant amount of work was still required for the mock up to appropriately represent the missing functions. Especially it needed to be improved to reflect the main concepts of UNITE, with regard to the integration of team members in the project office. For example, central to the UNITE concept is the personal context area, under full control of its owner, which provides the UNITE user with a private space to perform work outside any project space. However, this personal work area should also provide an overview/peripheral view on all projects that the owner is involved in, in particular notifications related to these projects. Secondly, the personal desk of a team member supports individual work within the context of a given project. When a personal desk is occupied, its owner can use all objects relevant to that context, and he or she has control over other team members that attempt to enter this personal desk. Thirdly, the collaboration area provides team members with a series of collaboration scenarios that they can use without specific permission from any of the other team members.

There had been a problem communicating some of these issues and their implications. The early mock up was criticised for not addressing these basic concepts, and in its representation of the working space as a 2D floor plan, which was found not to be meaningful. Rather it was felt that the user’s screen needed to be structured as a collection of different areas including ones to:

- indicate that a team member requests entering the user’s personal desk (e.g. in order to contact that user), and to enable the user to control entry to his or her personal desk;
- provide information on the team membership, independently from who is currently working in the team’s office;
- indicate who is working where, in an area at the top of the screen (as shown in Figure 1);
- enable navigation within the various contexts and sub-contexts available at the team’s office;
- provide details on the current context the user is in, e.g. on a collaboration scenario, on the work performed at a personal desk, etc.

Whilst it is obvious that a representation of these issues may not be most appropriately represented as a 2D virtual office, it is not clear what form of representation would in fact be most appropriate. In order to get the system up and working the functionality had to be represented in some way. It was appreciated that the mock up could not simulate every feature. However, all those features stated for inclusion in the Basic Platform had to be present so that they could be visualised by the users and the developers, and everyone could share the same view through a concrete representation of the user interface.

It was agreed that in order to achieve this common ground much closer interaction was required to clarify issues that were difficult to resolve with the semantic model as described in project reports. Closer interaction was also required between the developers to define the set of detailed interactions that needed to take place between each of the platform features and the user interface. Subsequently one of a series of developer’s workshop was organized to discuss implementation issues.

To summarise, the inspection of the emails shows that although problems were raised about the mock up and the SM, they were not immediately resolved. This is also noted in the responses to the questionnaire (see below). At this stage of the project neither email nor synchronous chat was used successfully to support in depth discussion of complex issues relating to the semantic model. The analysis of the minutes of the teleconference illustrated that the mock up was seen as having different roles and that these might be conflictual (e.g. the notion of an evolving interface based on
increased understanding, conflicted with the notion of it having to be an accurate representation of the SM).

**Semantic model and the mock up**

A questionnaire was distributed by email to project members to consider the role of the semantic model in relation to the mock up. Responses were received from the major stakeholders of the project i.e. user representatives, project manager, developer, evaluators and the interface developer.

The results are summarized below:

Most of the respondents thought that the mock up was helpful in clarifying issues relating to the Semantic Model. User representatives and the evaluators in particular saw it as their most accessible way of gaining insight into what the project was about. It therefore provided an opportunity for common ground to be achieved in the project which had not been possible through inspection of project documents or in meetings.

Clearly, the mock up was a focal point for discussions, usually because it did not adhere closely enough to Semantic Model. For example 'the earlier mock ups with avatars provided a better representation of the SM', although these were criticized by others for their inappropriate use of room metaphors. The mock ups ‘force discussion about the scope of the model which was undoubtedly valuable and deepened understanding.’ The mock up was seen as ‘an essential step, as it is not necessarily easy for everyone to discuss concepts in abstract terms only’. Through it the whole development team was brought into the discussion of the SM and misunderstandings became apparent.

So we would argue that the value of the mock up lay in its ability to uncover subtle difference in interpretation of the SM. Without the mock up the developers would not have achieved such a clear understanding of the SM. However, their reluctance to release any version of the mock up which did not closely adhere to the SM, or might have raised the users expectations seriously curtailed its opportunity for the evaluators and user representatives to have the same level of input at the conceptual stages of the project.

The early mock up’s lack of functionality meant that it was not possible at this point to assess the extent to which a system and interface architecture based on the semantic model is useful, so that the user representatives did not have a clear picture of the overall concept. It took a while to resolve the remaining conflicts of SM and mock up interface in a number of areas such as the users relationship with collaboration areas, the centrality and importance of collaboration on the interface (i.e. it is a major concept of UNITE but only a small part of the interface); the way in which people and their collaboration with others are represented on the interface. It was felt that when misunderstandings regarding the functionality of the system had arisen they were never fully resolved in email correspondence.

In summary it was important for the project to share the UNITE concept or vision. This has been variously articulated through the SM, mock up, meetings and project documents. All of these have played a role in articulating and bringing to life the UNITE vision for the different stakeholders in the project. However, it is believed that the mock up has played a crucial role in furthering common understanding amongst the team. This has been at the cost of other roles the mock up could have played in the project.
Discussion and recommendations

The need to establish quickly a common understanding of the system concept is central to many software development projects. Where the team are highly distributed and working in a modular fashion (see Figure 2), having inconsistent or partial understandings of the overall system may lead to costly delays, overspending of resources and lowering of satisfaction.

The UNITE project attempted to develop a common ground amongst project members by presenting the UNITE vision, firstly as a SM. However, there was either some reticence to discuss the implications and relationships within this model at the start of the project, or these had not been fully understood. Time was lost because misunderstandings of the SM led to a premature generation of code, which subsequently had to be revised and/or did not integrate with that in other modules (see Figure 2 for the importance of integration between modules).

The interface mock ups did provide a means of opening up the SM for those who could not fully understand the implications of the SM. However, the necessity for the mock up to adhere closely to the SM guidelines, and not raise user expectations by providing functionality which would not be supported in the final system, limited its usefulness as a means of providing early, iterative feedback from potential users and a wider discussion of the implications of the SM on real work. When the mock ups were produced they fulfilled most of their functions and were used by the team to understand the UNITE concepts.

The analysis of the emails and the questionnaire regarding the mock up clearly shows that communication in distributed, technology driven teams remains an issue. Emails providing comments are sent but there is no feedback as to whether they have been read or their contents acted upon. Additionally, asynchronous communication does not seem to be the best mode for discussing complex issues of this type – thereby necessitating additional, unscheduled meetings between sub groups at which real work is done.

In conclusion, the establishment of a common ground, with regard to the underlying concept of UNITE was hard to achieve, without the mock up it may well have been impossible. It may be supposed that distributed, software development projects of this nature may suffer from similar handicaps.

In terms of recommendations for the conduct of future research and development projects we would emphasise the need for a theoretical base, such as UNITE's semantic model to guide development. However, merely documenting this formally is clearly insufficient. Adequate time should be allotted at the start of the project to ensure that everyone has a clear overall picture of the project. It is obvious that long, technical documents are not reader friendly, and have a negative effect on the ability of people to take in new knowledge and concepts.

To facilitate this, and provide an equal opportunity for all project members to understand underlying concepts, many different representations may have to be produced, geared towards each member's needs. For example, system specifications drawn up by development team may be accurate, but unintelligible to those without technical training, who need to understand the basic concepts and what these mean for them at the level of 'doing work'. Figure 1, may be flawed in terms of its representation of the SM, but it provides a concrete, comprehensible example of most of the underlying concepts. In all projects there is a rush to coding, to meeting the first set of deadlines. This emphasis causes people to progress on the basis of incomplete understanding, whether it is a software or product design project.
Additionally, steps must be undertaken to facilitate, rather than inhibit discussion of underlying concepts, for example time should be allocated in meetings to discuss underlying concepts to ensure that there is a real common understanding, each members opinions should be received non-judgmentally and responses made to queries.

In summary, in the past the role of mock ups has been seen primarily in terms of facilitating user centred design, and in terms of generating early, iterative feedback to the developers (for example on functionality and usability). In this paper we have considered a different and somewhat overlooked role of the mock up in distributed projects, which is in their ability to provide a common ground and language for people to discuss and understand theoretical issues underpinning the project.

**Acknowledgements**
This project is funded by the European Union under the Information Society Technologies (IST) Programme (Project No. IST-2000-25436). This work was partially supported by the IST UNITE project; however, it represents the view of the authors only.
The authors would like to thank the members of the UNITE project for providing the information and material on which this was based.
References


