Trusty Interaction in Visual Environments

P. Bottoni[1*], *M.F. Costabile*[2*], *S. Levialdi*[1*], *M. Matera*[2*], *P. Mussio*[3*]

[1]Dipartimento di Scienze dell'Informazione, Università "La Sapienza", Roma, Italy
[2]Dipartimento di Informatica, Università di Bari, Italy
[3]Dipartimento di Elettronica per l'Automazione, Università di Brescia, Italy
[*]Pictorial Computing Laboratory, Università "La Sapienza", Roma, Italy

{bottoni, levialdi}@dsi.uniroma1.it, {costabile, matera}@di.uniba.it,
mussio@bsing.ing.unibs.it

Abstract. End-user computing requires that end-users trust the system and the results obtained by its use. The approach we have elaborated at the Pictorial Computing Laboratory of the University of Rome "La Sapienza" is the result of several experiments through the years in design and use of end-user visual computing systems and is aimed at improving the system *trustworthiness*. To this end, our approach adopts the notation developed by the users in their working environment as the kernel for the Visual Language used during human-computer interaction, supports users while navigating in the virtual space by establishing a system of cornerstones, within a scaffold, and provides control on the system to trap user slips and errors. The paper reports and discusses some results from our experience in the design and use of end-users visual environments.

1. INTRODUCTION

The emerging *Information Society* requires the development of computer systems and services that provide easy access and quality in use to all possible end-users. Brancheau and Brown define *end-user computing* as "… the adoption and use of information technology by personnel outside the information system department, to develop software applications in support of organizational tasks" [Brancheau 93]. End-users (or *users* for short) increasingly are people not expert in computer science, who use interactive computer environments to perform tasks of which they are responsible. Often, accomplishing such tasks requires the users to program system functionalities or at least executing activities similar to programming. Users are responsible for the activities accomplished through the system and for the produced results. Therefore, end-user computing requires that end-users trust the system and the results obtained by its use. Moreover, they must always understand the consequences of the system activity with respect to the execution of their tasks, and be in control of the interactive computation, without getting lost in the virtual space. Also, the system has to trap the users' errors and maintain itself viable, i.e., maintain itself in a predictable set of states, in which it never crashes.

It is critical for users to accomplish their tasks in an economic (implying minimal user action and minimal memory load) and reliable way: users accept the support of automatic tools to perform their tasks only if they can trust such tools and the results obtained through them. To this end, it is critical for the designer to get a closer mapping between the real world, in which users operate, and the designed virtual world that must support the users in their work [Green 96].

This paper argues that for all these reasons the attitude engendered into users when working in interactive visual environments is crucial for system acceptance. This hypothesis stems from experience in the design and use of visual environments, which exploits the approach to visual environment design we developed at the Pictorial Computing Laboratory (PCL) of the

University of Rome "La Sapienza".
The PCL approach is aimed at the design of visual environments that are accepted with satisfaction by their users. To this end, it recognizes that in an interaction process *two semantics* exist, one relative to the user and the other implemented within the system; the designer's goal is to make them as close as possible, thus reaching *adequate human-computer communication* (i.e., the human and the computer associate a similar meaning to a same message) [Chang 96] and *system communicability* [Prates 00], which are prerequisites to the system acceptability by the intended users. The *users' notations*, developed by the users in their working environments, are the main elements of the users' communication language in which they express their culture. Therefore, in order to reach adequate communication the designers have to capitalize on these notations, adopting them as the kernel for designing a visual environment; they also evolve and augment the notations to cope with new possibilities offered by computer-driven reasoning. This approach especially enforces the system *trustworthiness*, which in our opinion is a *necessary* dimension for system acceptability, to be taken into account during the design of visual environments. This position stems from the PCL experience in designing *critical* end-user systems, i.e., systems supporting end-users in achieving tasks in which every mistake or slip of the user, or every failure of the system has a valuable cost, and whose results must be reliable.
Nowadays, in most cases, users interact with information systems via screen based devices. For this reason, the discussion is here restricted to WIMP interaction. In this case, according to the PCL approach, each message on the screen can be described as a visual sentence, i.e., an element of an *Interaction Visual Language*, as illustrated in [Bottoni 00].
In the rest of the paper we first introduce in Section 2 the PCL view of Visual Human-Computer Interaction. Then we discuss how the issues deriving from the PCL approach impact the system trustworthiness. Such concepts will be illustrated through examples drawn from our experience in the design of visual environments for healthcare professionals and for mechanical engineers. More specifically, Section 3 presents the conditions for trusty interaction. Section 4 discusses how our approach satisfies classical usability requirements. Section 5 shows how further requirements can be derived from the model and the formal definitions we propose. The last section draws the conclusions and indicates some open issues.

2. A VIEW ON INTERACTION THROUGH VISUAL SENTENCES

The PCL adopts what Preece et al. call a *holistic* approach to visual environment design [Preece 94]. In the design process, the decisions about the way in which the user interface should look like and how it should behave are taken depending on how this will be physically communicated to users, and attention is focused both on the appearance and the behavior of the interface. However, differently from other holistic approaches, the PCL approach adopts a formal technique to specify the computational meaning of what is progressively defined [Bottoni 99]. In this way, a conceptual model of the visual environment is incrementally built during the design, and can be displayed to users as a system prototype and validated by usability evaluation techniques, while contextually it can be verified by formal techniques.
Human-Computer Interaction is a process based on visual communication between two participants, namely the human user and the computer: they communicate by materializing and interpreting a sequence of messages at successive instants of time $t_1,...,t_n$, the human using his/her cognitive criteria, the computer using the criteria programmed by the designer [Bottoni 99]. Hence, the visual environment itself is considered as a generator of messages from the designer to the system user [Prates 00].
Two semantics are always implicitly defined in any interaction: one internal to the visual

environment, in which each message is associated with a computational meaning, as defined

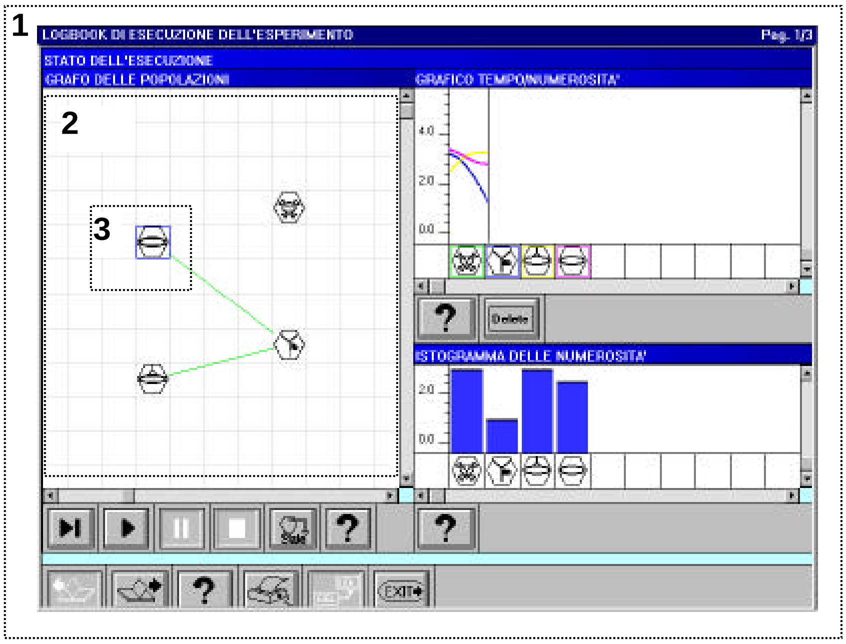

1
LOGBOOK DI ESECUZIONE DELL'ESPERIMENTO
Pag. 1/3
STATO DELL'ESECUZIONE
GRAFO DELLE POPOLAZIONI
GRAFICO TEMPO/NUMEROSITA'
2
3
4.0
2.0
0.0
Delete
ISTOGRAMMA DELLE NUMEROSITA'
2.0
0.0
?
EXIT

Figure 1. An image displayed during the Human Immune System simulation through the VIPERA visual environment [Bianchi 99]. Three different css are framed by dotted lines.

by the designer and implemented in the visual environment, and one proper to the user performing the task, depending on his/her role in the task, as well as on his/her culture, experience, etc. As observed in [Chang 96], the interactive accomplishment of the user task requires that a similar meaning be associated by the user and the visual environment with each message, i.e. that an *adequate communication* be reached. The goal of a successful design is to bring the system semantics to reflect the user's one, so that the messages exchanged during the interaction are properly understood by the user and adequately managed by the computer.

In the case of two-dimensional direct manipulation, on which this paper focuses, the exchanged messages are the whole images represented on the screen, formed by text, graphs, pictures, icons, etc. Humans interpret such images by recognizing *characteristic structures* (css or structures for short), i.e. sets of image pixels which humans recognize as functional or perceptual units. The cs recognition results into the association of a meaning with a structure. Humans express the *meaning* attributed to the cs by a *verbal description*. Such an identification of css is influenced by the (dis)similarity with graphical entities and constructs traditionally adopted in the user's community.

Figure 1 shows a typical screen display generated during an interactive simulation with VIPERA [Bianchi 99]. VIPERA (*VIsual Programming Environment foR evolving Agents*) is a visual environment designed to support the research, clinical, and teaching activities of immunologists - a community of physicians who study the human immune system and treat its diseases. Immunology researchers and clinicians, as well as their students, are the users VIPERA has been designed for. They share a common basic culture on the human immune system, expressed through notations which are not formalised in the computer science sense. Through these notations they express and communicate in their literature the immunological models and results.

In the example in Figure 1, the immunologist is simulating the behaviour of a population of biological entities. The image on the display is composed of several css in the form of texts, graphics, pictures, etc. An immunologist looking at the image recognises in each hexagon a cs representing a population of biological entities, because s/he recognises the icon inside each hexagon according to the rules adopted within the immunologists' community. For example, the icon within frame 3 represents the population of Bacteria, which can be described by enumerating the types of elements present in the current situation and the number of elements for each type. The verbal description is therefore summarised by a set of names of types and a set of numbers.

For each recognized population, the immunologist associates each entry in the histogram in the bottom right window with the cardinality of the population, as indicated by the underlying icon. S/He also associates each plot in the graphs at the top right window with a population history by color. Note however that also the graph within frame 2 -in which the hexagons representing individual populations are nodes, and edges denote the existence of a communication between two populations- constitutes a cs on its own, whose description summarises the overall state of the patient immune system.

It is worth noting that the immunologists derive the meaning of some css from non-biological sources of knowledge. For example, they exploit their experience with video recorders, other electronic appliances, and direct manipulation interfaces to interpret the buttons at the bottom of the windows as the mechanisms to send commands to VIPERA. Moreover, some textual css help them understand the meaning of interface widgets.

Note that css of the last two types are also understandable by users who are not immunologists. Such users can for example understand the meaning of the buttons and of the text and deduce that they are in presence of an animated document, related to some simulation of the human immune system.

The whole image in Figure 1 appears as a document for a user, who understands it according to his/her role in the task, culture, experience, and social relations in the work environment. Such a document can also be considered a cs itself. The user infers the meaning of this cs from the meanings of the elementary css, composing them according to the rules of his/her own language.

From the user point of view, communication with the application exists only via the user's perception of the rendered presentation of the message. This implies that the user should be able to create a mental model of the application process based on the provided presentation. On the other hand, the visual environment associates graphical entities with computational constructs as well. It is exactly this association which makes the computer able to interpret the captured user actions (such as clicking on a button) with respect to the image on the screen, possibly firing computational activities whose results are materialised on the screen via creation, deletion, or modification of css.

The problem of achieving adequate communication, therefore, requires that a precise correspondence be defined between the structures perceived by the user and those foreseen by the designer and implemented in the visual environment. In general, this requires the formal definition of those arrangements of pixels which have to be considered as css, and of the association between such structures and their perceived or intended meaning. To this end, we have introduced the notion of *characteristic pattern*.

We already said that a characteristic structure cs is a set of pixels, perceivable on the screen. A pixel within a digital image is formally described as a triple (row, column, value). The meaning associated with a cs is formalised as an *attributed symbol*, called u, consisting of a type name - the *symbol* - and a tuple of properties - the values of the attributes providing a description and interpretation of the cs. The association between a cs and its description u is expressed by two functions, intcp and matcp. The *interpretation function*, intcp, associates the cs with its description u. The *materialisation function*, matcp, associates the attributed symbol u with the cs. A cs, the attributed symbol u, and the pair intcp and matcp constitute a *characteristic pattern*(cp), i.e. cp=<cs, u, <intcp, matcp>>.

In an image i, several css can be identified. These css became cps when a description u is associated to each cs. For example, in Figure 1 the graph in frame 2 is a cs, whose description summarises the state of the patient and maintains references to the descriptions of the individual populations (hexagon). The existence of such contextual values organises the populations in a "part-of" relation. Moreover, i as a whole can be associated with a symbol u_i synthesising the overall properties and the global meaning of the image i. In any case, the set d of all the attributed symbols appearing in the cps composing i constitutes a description of the image i.

Two functions, int and mat, can be defined on the basis of the individual functions in the cps. The relationship between the image i and its semantics is summarised by the triple: vs=<i,d,<int, mat>>. We call vs *visual sentence*, i image component and d description component of vs. A set of vss is a Visual Language (VL).

A formal theory that provides a finite definition of VLs by a special family of rewriting systems, the *Visual Conditional Attributed Rewriting Systems* (*vCARWs*), has been described in previous papers [Bottoni 99, Bottoni 97, Bottoni 98b]. Moreover, a special type of vCARW has been defined to model the transformations occurring among vss in an interactive process. Very briefly, an interactive process can be specified by the possibly infinite set of all the sequences of vss that, starting from an initial vs, vs_0, are determined by the sequences of user actions and system computations in the process [Bottoni 98b]. Each sequence in the set describes a specific user-computer interaction session. The set of all sequences of vss that can be generated from vs_0 constitutes the *Interaction Visual Language* (IVL). For example, Figure 1 represents the image component of a vs in the IVL of the VIPERA system.

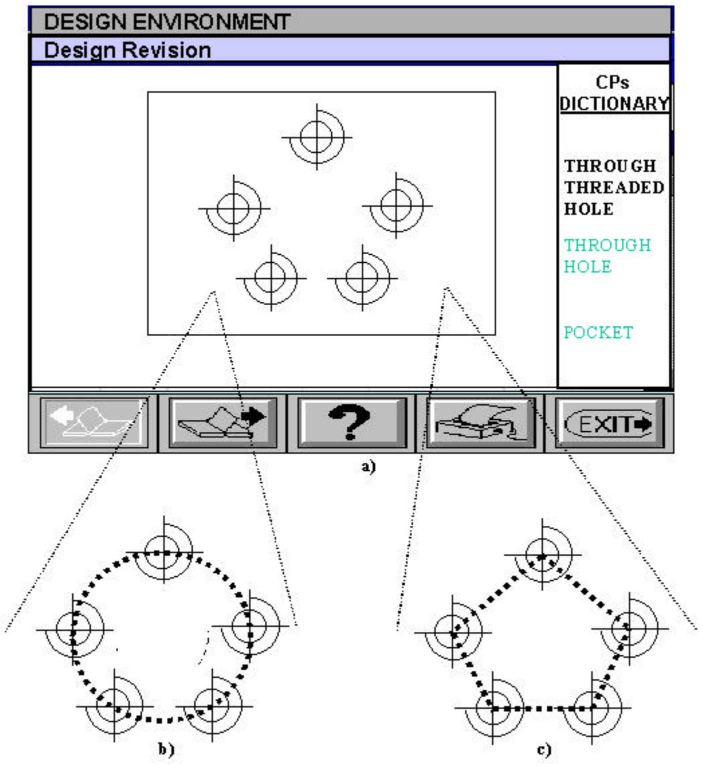
A designer needs to define the IVL by finite means, in a way analogous to the grammar definition used for traditional programming languages. The IVL definition by finite means requires the definition of: a) the set of admissible user actions and the set of algorithms by which the system executes the required computations, b) how to represent these actions in a graphical way; c) how to relate user actions and system computations with vss, i.e., how they determine a transformation of a vs into a different one, and d) how to generate the set of sequences of vss in the IVL. This last point is achieved by first specifying how to generate a vs from another and then specifying how to apply repeatedly such a transformation. The process of transforming a vs into a second one can be formally specified by defining a finite set of rules *P*, guarded by the execution of a human action, and a rewriting relation. The rewriting relation states how to apply a rule in *P* to transform a vs into a different one. This corresponds to the definition of a special type of *vCARW*, the *Enabling Visual Conditional Attributed Rewriting Systems* (*evCARWs*) [Bottoni 98a]. In this paper, we will not describe further this point, rather we will focus on the IVL design for trusty interaction.

3. TOWARD TRUSTY INTERACTION

As mentioned in the introduction, system *trustworthiness* is in our opinion a necessary dimension for the system acceptability to be taken into account during the design of visual environments. An interaction is trusty if the users understand each situation (i.e., they associate the correct meaning to each image component of a vs in the IVL), and do not get lost in the virtual space. Two necessary conditions for trusty interaction are: 1) reaching an adequate level of communication, and 2) providing facilities to orient the user when navigating in the virtual space. Sections 3.1 and 3.2 discuss these two conditions.

3.1. Reaching Adequate Communication

The recognition of the existence of two semantics is the starting point for reaching an adequate communication between human and computer. Indeed, adequate communication occurs when the human and the computer associate a similar meaning to a same message (or part of it) [Chang 96]. In order to identify the users' semantics, we recommend to exploit the users' notation as the kernel of the definition of the IVL through which human and computer



communicate. Users' notation embeds context, task and procedural knowledge possessed by

Figure 2. A sketch from a mechanical design environment: in a) a technical drawing in which five top views of threaded hole appear; in b) and in c) possible patterns organizing the five threaded holes in two different tool paths emerging in the production engineer's mind.

the users. It embeds knowledge both explicitly and implicitly: explicitly in symbols and rules used for constructing the users' documents; implicitly, because the shape and the spatial arrangements of symbols allow users expert in the application domain to recognize structures and relations meaningful for their tasks, but too vaguely defined to be made explicit in a declarative format.

A typical example of how implicit knowledge is used in traditional environments occurs when users, experts in some specific fields, communicate and reason through documents. Different users, with different roles, interpret a document to reach different goals. In these cases, the document is built from a set of symbols and according to a set of rules accepted by all the users. However, due to their different roles and goals, it may occur that users with different roles and responsibilities in the task perceive different structures emerge from the document at hand which were not foreseen by the document producer. For example, consider a production engineer looking at the technical drawing of Figure 2a, which was produced by a designer. This drawing represents five css, each one representing for an engineer a specific cp - the top view of a threaded hole [Mussio 94]. The drawing was built according to the rules codified by the mechanical engineering community. However, the designer has placed the top views of threaded holes, taking into account the mechanical properties of the part to be produced. But the production engineer reasons on the document to derive the best path an instrument has to follow to produce the mechanical part. In his/her mind the structures in the drawing cooperate to form a structure in the image that is not actually traced in the image, as shown in Figure 2 b) and c).

Different interpretations may also occur when different users read electronic documents - i.e. the image component of a vs. When the message is expressed in a form resembling the traditional user notation, users actually see more than what is formally specified in the computer programs [Bianchi 99, Mussio 91]. Their culture and skill make them recognize unforeseen patterns both in the vs and in the vs sequences generated during interaction.

The adoption of users' notations as the kernel of the IVL is a fundamental step to make the interaction process understandable and checkable by the users: users may justify results expressed in their notations on the basis of their experience and not on the basis of an algorithmic explanation. In this way, the adoption of user notation facilitates the reaching of the closeness of mapping between the real world, in which users operate, and the virtual world which supports their work. However, as pointed out by Mayhew [Mayhew 92], the mere adoption of this notation may bring to under-use the system. In fact, the users' notation has the advantage of being completely familiar to the user, but also the disadvantage of having been defined without taking into account the existence of computing systems. The PCL approach therefore proposes to augment and adapt the original notation to fully exploit the computing capability of the interactive systems. In the augmented language, symbols are able to show their state, for example, assuming a color to show that they have been selected, and can be associated to a specific functionality to favor the interaction with the user.

3.2. Facilities to Orient User Navigation

In order to provide support during interaction and navigation in the virtual space, orientation cornerstones must allow users to recognise rather than recall, i.e., without interpretation effort, the current situation and the path they are following for solving their problem, i.e. where they are, where they can go, etc. Similarly, the facilities to navigate in the identified direction must be recognised and used at a low cognitive cost. In order to orient the user navigation, the PCL approach proposes the specification of *scaffolds* and *frame*s in the design of the vss to be displayed on the screen during interaction.

The *scaffold* is informally defined as a set of css in a vs that facilitates the understanding of the overall strategy for performing a task and recording the history of the interaction. It consists of a set of icons, text lines and widgets used to denote:

a) the *activities* that can be performed, such as select a cs, launch a program, terminate the current activity, save current results;
b) the *cornerstones*, which allow users to get oriented during the task execution. Examples of cornerstones are window titles, status bars, etc.

Given a set of images (the image components of vss), the *frame* for that set consists of all the css which are maintained unchanged (except for a few admitted transformations such as positive/negative inversion) in all the images. If the set is formed by the image components of vss that constitute the steps in an interactive session, then the frame provides a constant background against which to observe the evolution of the human-computer interaction.

During an interactive session, the visual sentence whose image is on the screen describes the state of the program in control of the system. In particular, the image component is maintained in the bitmap of the system, while the description component includes both the

data structure and the programs to be executed in reaction to users' actions. What the users see on the screen is therefore the materialisation of a view (projection) of the state. It is often the case that during the interaction a same vs is produced several times, as an effect of different actions performed on different vss. In particular, sequences of actions may produce a closed path between two vss (e.g. a sequence of undo and redo actions). In general some css in the image component of a vs remain unchanged, thus identifying the frame in which the interaction takes place.
The image components of the vss produced during the interaction must be designed so as to

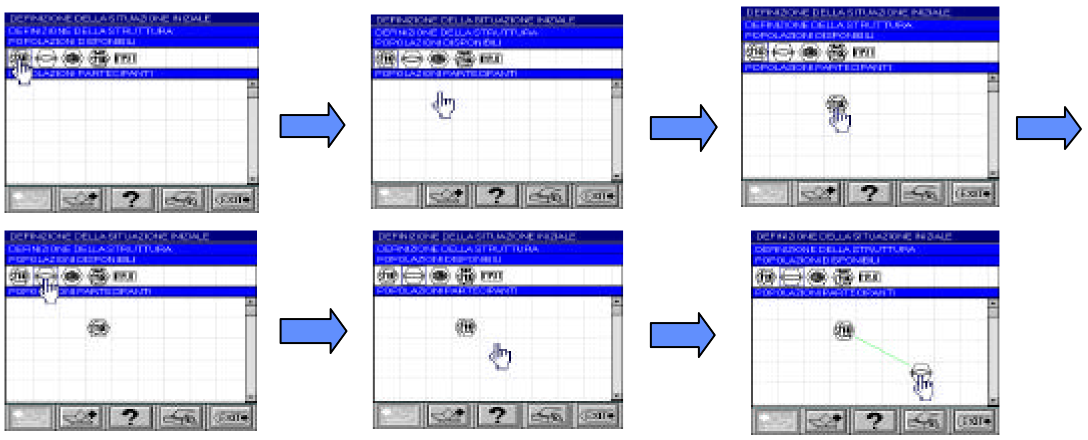

DEFINIZIONE DELLA SITUAZIONE INIZIALE
DEFINIZIONE DELLA STRUTTURA
POPOLAZIONI DISPONIBILI
POPOLAZIONI PARTECIPANTI
?

Figure 3. The sequence of vss for the interactive definition of the initial state of an experiment in the VIPERA system.

provide users with the data necessary to understand the current state of the system and the actions that can be performed. This means that special attention must be devoted to the design of both scaffold and frame. An example of an element of a scaffold is the menu bar in a Macintosh application. Indeed, it includes menus that represent some activities that can be performed. Some variations in these menus are also admitted, such as the negative/positive appearance of the label of the menu item and its background. As examples of scaffold and frame in the VIPERA system, let us consider Figure 3, which shows a sequence of vss for the definition of the initial state of an experiment. The user has moved the cursor on the first icon on the top left of the work area and clicks in order to select it. Note that the cursor is another important cs in this vs, and with its shape and position provides useful feedback to the user. The user then moves the cursor in a place of the work area where s/he wants to place the element represented by the selected icon (second vs). The user clicks in this position to have the element put there (third vs). Now, the user wants to place another element in the work area, connected to the previous one. For this, s/he selects the element by clicking on the corresponding icon (fourth vs), then moves the cursor to a position in the work area (fifth vs), and by clicking at this position the selected element is displayed with an arc that connects it to the previous element (sixth vs). Figure 4a shows the scaffold for the sequence of interaction steps shown in Figure 3. The scaffold consists here of all icons, text lines and other widgets used to denote the *cornerstones* and the *activities* that can be performed by the user. Therefore, all css in the vss, except the cursor cs and those css appearing in the working window as result of the user interaction, belong to the scaffold. Figures 4b and 4c denote the frames of the vss 1-3 and 1-6 respectively. Indeed, the frame consists of all the css that are maintained unchanged in all the image components of the vss in a sub-sequence. Therefore, the frame for the sub-sequence 1-3 is slightly different from the frame for the sub-sequence 1-6 and, as shown in Figures 4b and 4c, is formed by all css in the vss excluding those in the areas covered by the clouds.

4. HOW THE PCL APPROACH MATCHES USABILITY PRINCIPLES

The HCI literature has provided some principles and guidelines for the designers in order to create usable interactive systems. They provide the designer with a basis for making good decisions. Among the most popular ones are Nielsen's usability principles [Nielsen 93] and Shneiderman's golden rules [Shneiderman 98]. The two sets of rules are very much correlated, and we focus our attention on some of the Nielsen's ones for comparing the principles underlying the PCL approach, primarily devoted to designing visual environments,

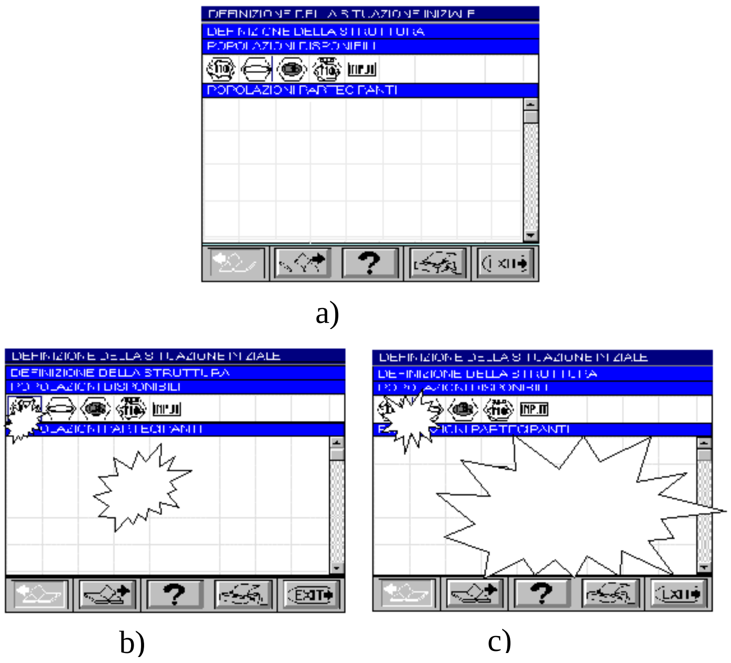

DEFINIZIONE DELLA SITUAZIONE INIZIALE
DEFINIZIONE DELLA STRUTTURA
POPOLAZIONI DISPONIBILI
POPOLAZIONI PARTECIPANTI
?
EXIT
a)
b)
c)

Figure 4. Scaffold and frames for the definition of the experiment initial state in the VIPERA system: a) the scaffold for the six vss in Figure 3; b) the frame for vss 1-3 in Figure 3; c) the frame for vss 1-6 in Figure 3.

with these traditional principles for designing user interfaces of any style. However, as a further contribution towards the design of interactive environments accepted with satisfaction by their users, the PCL approach stresses system *trustworthiness*, and provides indications on how to design facilities to orient the user when navigating in the virtual space and how to reach adequate communication.

The first Nielsen's principle is "Simple and natural dialogue": it states that dialogues should not contain information that is irrelevant and rarely needed, and all information should appear in a natural and logical order. The second principle is even more explicit saying "speak the user language", i.e., the dialogue should be expressed clearly in words, graphics and concepts familiar to the users. The main principle of our design methodology actually demands to capitalize on the users' notation expressing their culture. This is the analogous of "speak the user language"; a great part of our work concentrates on observing the users in their working

environment, identifying the notation they adopt for communicating in the real world and for producing documents during their daily activities, and formalizing it in a specific VL, whose image part still allows its users to exploit their implicit knowledge. As far as the first principle is concerned, in order to generate simple and natural dialogues, it is necessary to analyze the task model, the way users do things. Accordingly, we have stressed in our design methodology the importance of starting from user and task analysis, properly augmenting it in order to take care of the computer capabilities, and abstracting from the observation of the produced documents and of the users' activities the definition of the kernel of an IVL.
The third Nielsen's principle is "Minimise user memory load": the user should not have to remember information from one part of the dialogue to another. In our approach, each visual sentence in the sequence appearing on the screen during human-computer interaction is constructed so as to show the current system state, and the actions the user may perform are visible and represented by appropriate css; moreover, the automaton governing the interaction, derived from the formal definition of the vCARW, makes visible in the current visual sentence only those css corresponding to legal actions for the user. In this way, not only do we follow the above principle, but we go further, avoiding to the user any possibility of performing incorrect actions that are immediately trapped by the system. This is in line with Nielsen's principle "Prevent errors".
Other two basic principles are "consistency" and "feedback". Consistency in the user interface has to be maintained at various levels: for example, layout of widgets in different screens has to be consistent, in order to help users to easily find out the desired widget at any time; a same operation available in different screens of the interface must be represented by the same widget, and so on. Our approach of deriving the IVL from the user traditional notations clearly favours consistency in icon design. We also stress the feedback principle, since it is a way to keep the user in control of the interaction, by providing a visual indication of any action performed by the user or any operation performed by the system. To enforce the principle that the user must be always in control of the interaction, we reiterate that we operate so that in any interface screen (vs) the actions the user may perform be visible and represented by appropriate css; moreover, at any time the user can interrupt the interaction session by acting on an appropriate interface widget, thus conforming to Nielsen's principle "Clearly marked exit".

5. FURTHER ISSUES IN DESIGNING TRUSTY VISUAL ENVIRONMENTS

The PCL approach proposes a model of Visual HCI and a formal methodology for specifying a visual environment [Bottoni 99], which support the precise definition of further properties with respect to usability, concurring to obtain a trusty interaction and system acceptability by the users. These precise definitions are the basis for the verification and validation of the system [Dix 98]. The identified properties are:

1. *Non-ambiguity in interpretation.* Ambiguity arises when one of the two communicants - the human or the computer- associates two or more different meanings to a same message, and may flip from one meaning to the other during a same reasoning process. Since the visual environment is formally specified, some suitable conditions can be derived on the formal definition of the IVL to avoid ambiguity.
2. *Adequate communication.* Equivocal situations arise when the two communicants associate two different meanings to a same message but are not aware of this fact. Misunderstandings arise due to a different interpretation of a same message by the human and the visual environment. While the visual environment interpretation is formally defined, the human interpretation depends on culturally and socially situated factors. For this reason, absence of misunderstandings cannot be formally verified [Tondl 81], rather

it can only be experimentally validated.

3. *Deterministic interaction.* Each user action always results in a same visual environment reaction, if executed in a same visual sentence. This property can be formally defined as a constraint on the formal IVL definition.
4. *Viability of the visual environment.* Every sequence of user actions maintains the visual environment viable, i.e., in a predictable set of states, in which the visual environment never crashes. The PCL model supports the definition of a control automaton to trap users mistakes and slips. This automaton specifies how the visual environment has to govern the interaction, making active in the current visual sentence only those css corresponding to legal actions for the user, and trapping the users' incorrect actions [Bottoni 99].

6. SOME OPEN PROBLEMS

The adoption of the user's notation seems to facilitate the reaching of adequate communication at the cognitive level. In fact, users recognize visual entities on the screen (both textual and graphical) according to a metaphor familiar to them. However, it leaves open some problems at the articulatory level, since users are required to use gestures to interact with such entities, which are not conform to the metaphor. For example, as reported in Figure 3, in order to establish the initial state of a simulation experiment in the VIPERA system, the immunologist creates an image which is a metaphoric representation of what s/he sees through the microscope (the set of the cell populations), but the gestures by which s/he creates the image have no resemblance with the real one. Such a difference makes it difficult to the user to think at a high level of abstraction, because they have to concentrate on unfamiliar sequence of gestures [Bordegoni 00].

In general, the css in the IVL suggest gestures that can only be partially executed:

1. virtual sliders and buttons are managed through gestures which resembles those performed in the real world to manage real sliders and buttons;
2. brush, knobs and leverages require a different articulatory strategy;
3. other specialized widgets require articulation which may be unrelated to real world operations (cell selection, part mounting, establishing an electrical connection).

The haptic feedback for the widgets defined in case 2. and 3. is always different in the real and the virtual world, even when the visual feedback is in accordance with the metaphor exploited in the virtual world. Therefore, one open problem is to obtain a closer mapping between the real world (in which the users operate) and the designed virtual world, which has to support users when performing operations. To this end it is necessary to study new technologies to reach conformance of gestures in real and in a virtual environment, or at least to reduce the cognitive cost associated to the articulatory distance.

ACKNOWLEDGEMENTS

This work is partially supported by the EC grant FAIRWIS, project no. IST-1999-12641.

REFERENCES

[Bianchi 99] A. Bianchi, M. D'Enza, M. Matera, A. Betta, Designing Usable Visual Languages: the Case of Immune System Studies, in *IEEE Symposium Visual Languages'99*, Tokyo, Japan, September 1999, pp. 254-261.

[Bordegoni 00] M. Bordegoni, U. Cugini, P. Mussio, M. Matera, The Role of Continuity in Haptic Interaction Systems, in *CHI2000 Workshop on "Continuity in HCI"*, The Hague, The Netherlands, April 2000.

[Bottoni 97] P. Bottoni, M.F. Costabile, S. Levialdi, P. Mussio, Defining Visual Languages for Interactive Computing, *IEEE Transactions on Systems, Man and Cybernetics-A*, 27(6), 1997, pp. 773-783.

[Bottoni 98a] P. Bottoni, S.K. Chang, M.F. Costabile, S. Levialdi, P. Mussio, Dynamic Visual Languages, in *IEEE Symposium Visual Languages'98*, Halifax, Nova Scotia, Canada, September 1998, pp. 14-21.

[Bottoni 98b] P. Bottoni, M.F. Costabile, S. Levialdi, P. Mussio, Specifying Dialog Control in Visual Interactive Systems, *Journal of Visual Languages and Computing*, 9(5), 1998, pp. 535-564.

[Bottoni 99] P. Bottoni, M.F. Costabile, P. Mussio, Specification and Dialogue Control of Visual Interaction through Visual Rewriting Systems, *ACM TOPLAS*, 21(6), 1999, pp. 1077-1136.

[Bottoni 00] P. Bottoni, M.F. Costabile, S. Levialdi, M. Matera, P. Mussio, Principled Design of Visual Languages for Interaction, in *IEEE Symposium Visual Languages 2000*, Seattle, Washington, USA, September 2000, pp. 145-152.

[Brancheau 93] J.C. Brancheau, C.V. Brown, The Management of End-User Computing: Status and Directions, *ACM Computing Surveys*, 25(4), 1993.

[Chang 96] S.K. Chang, P. Mussio, Customized Visual Language Design, in *Eighth Int'l Conference on Software Engineering and Knowledge Engineering*, Lake Tahoe, Nevada, USA, June 1996, pp. 553-562.

[Dix 98] A. Dix, J. Finlay, G. Abowd, R. Beale, *Human Computer Interaction*, Prentice Hall, London, 1998.

[Green 96] T.R.G. Green, M. Petre, Usability Analysis of Visual Programming Environments, *Journal of Visual Language and Computing*, 7(2), 1996, pp. 131-174.

[Mayhew 92] D.J. Mayhew, *Principles and Guidelines in Software User Interface Design*, Prentice Hall, Englewood Cliff, 1992.

[Mussio 91] P. Mussio, M. Pietrogrande, M. Protti, Simulation of Hepatological Models: A Study in Visual Interactive Exploration of Scientific Problems, *Journal of Visual Language and Computing*, 2, 1991, pp. 75-95.

[Mussio 94] P. Mussio, On Positive Use of Ambiguity in Machine Vision and Image Understanding, *Human and Machine Vision. Analogies and Divergencies*, V. Cantoni (ed.), Plenum Press, NY, 1994.

[Nielsen 93] J. Nielsen, *Usability Engineering*, Academic Press, NY, 1993.

[Prates 00] R. Prates, C. De Souza, S. Barboza, A Method for Evaluating the Communicability of User Interfaces, *Interactions*, 7(1), 2000, pp. 31-38.

[Preece 94] J. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland, T. Carey, *Human-Computer Interaction*, Addison-Wesley, 1994.

[Shneiderman 98] B. Shneiderman, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Addison-Wesley, NY, 1998.

[Tondl 81] L. Tondl, *Problems of Semantics*, Reidel, 1981.