



HAL
open science

Design area for assistance to maintenance based on augmented reality.

Nadia Zenati, Nouredine Zerhouni, Karim Achour

► **To cite this version:**

Nadia Zenati, Nouredine Zerhouni, Karim Achour. Design area for assistance to maintenance based on augmented reality.. IEEE International Conference on Industrial Technology, ICIT'06., Dec 2006, Numbai, India. pp.2960-2965, 10.1109/ICIT.2006.372623 . hal-00341113

HAL Id: hal-00341113

<https://hal.science/hal-00341113>

Submitted on 24 Nov 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Design Area for Assistance to Maintenance based on Augmented Reality

N.Zenati^[1], N.Zerhouni^[2], K.Achour^[1]

[1]Advanced Technology Development Centre, [2] Automatic Laboratory of Besançon
nzenati@cdta.dz, zerhouni@ens2m.fr, kachour@cdta.dz

Abstract: One of the main weaknesses with traditional computing is the fact that the numerical world of the computer is decoupled from the user's real world. The application of Augmented Reality (AR) can provide interactive systems in which real objects and computer data are combined in a cohesive way. This new paradigm has many potential applications in various fields, in particular in the maintenance domain. It allows the user to see computer generated virtual objects superimposed to the real world through the see-through Head Mounted Display (HMD). The technician of maintenance, when using this system, can interact with the virtual world and have additional information, such as instruction for performing maintenance tasks in form of text messages, images, 3-D models of pieces or audio such as speech instruction.

In this paper, we propose a design process of the maintenance system focused on the analysis of the interaction between the user, the system and the real world. This area is based on the UML notation. The use of UML represents our ergonomic and software design process basis for AR systems. This process also is based on ergonomic characteristics study within a UML system description and on the hybrid PAC-Amodeus model architecture adaptation for the AR systems.

I. INTRODUCTION

Augmented Reality (short: AR) is a new way of human-computer-interaction, where virtual objects are added to real scenes provided by a video camera in real time [10] [11]. They are inserted in the right positions and complement the real picture. The digital information merges with the user's environment so that the user can perceive currently important information directly where it is needed.

An AR system that can enhance a user's view of the surrounding scene with annotations on the scene content, has many potential applications; These include medicine [2] [8], maintenance, repair and training in industrial environment [5] [9] [13] [14] [15], design [17], robotics and tele-operation [4], games [16], etc. An overview is given in Fig.1.

AR is derived from virtual reality (VR) in which the user is completely immersed in an artificial world. In VR systems, there is no way for the user to interact with objects in the real world. Using AR technology, user can thus interact with a mixed virtual and real world in a natural way.

This new paradigm has many potential for a large number of useful applications, in particular in the maintenance domain that has received considerable interest. It allows the user to see computer generated virtual objects superimposed to the real world through the see-through Head Mounted Display (HMD).

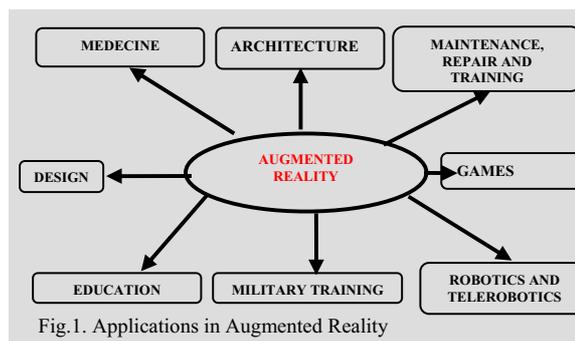


Fig.1. Applications in Augmented Reality

The technician of maintenance, when using this system, can interact with the virtual world and have additional information, such as instruction for performing maintenance tasks in form of text messages, images, 3-D models of pieces or audio such as speech instruction.

Augmented reality constitutes a very significant subject of study in the IHM field. Recent work undertaken aims at supporting the interaction with the user physical environment, i.e. the real world. This first comparison could inform us on the AR paradigm at these beginnings like its limitations. However, the diversity of systems claiming as AR systems and in particular those belonging to the maintenance field requires a quite precise framework of definition and design.

In this paper, we propose a design process of the maintenance system focused on the analysis of the interaction between the user, the system and the real world. This area is based on the UML notation. The use of UML represents our ergonomic and software design process basis for AR systems. This process is also based on ergonomic characteristics study within a UML system description and on the PAC-Amodeus model architecture adaptation for the AR systems.

II. DESIGN STEPS OF THE INTERACTIVE SYSTEM

Augmented Reality (AR) is a new paradigm which is used to describe a system, which enhances the real world by superimposing computer-generated information on top of it.

In the maintenance field, the existing design methods of a system based on AR require especially taking into account the physical world around the system user. The approach we have adopted is based on the ergonomic properties of the IHM (Man Machine Interface).

A development process of an information processing system requires a structural environment which manages and supervises all the design and specifies the key steps. Various models were elaborated to satisfy the development requirements. Each of them has his own specificities. The most used models are the cascade model, the spiral model and the V-model [12]. The V-model, although simplistic, has the advantage of specifying the key steps of any development process. For this reason, we use it as support and methodology in our application. The V-model adapted distinguishes four principle stages: the requirement specification, the design, the coding and the tests (see Fig.2). The particularity of Augmented Reality systems lies in the duality of the user interaction with the system. Indeed, it's composed of two facets: one with virtual entities and the other with real entities as the handled real tools. The use of Augmented Reality systems must, thus, be declined according to properties established for an interactive system [7], but it must also consider the two interaction facets in order to guarantee its harmonious integration of it. For this purpose, the adopted approach to study the utilizability of Augmented Reality system is based on the IHM ergonomic properties.

In particular, a significant point of the IHM design consists in making the pertinent data of the user task realization perceptible and this in a correct and easily interpretable way. Outputs of the systems consequently constitute the focus of our analysis. Among the existing ergonomic properties, two are closely related to outputs of interactive systems, namely *observability* and *honesty* [1] [6] [7].

- 1) *Observability*, characterizes the ability of the user to evaluate the internal state of the system from its perceivable representation.
- 2) *Honesty*, characterizes the ability of the system to ensure that the user will correctly interpret perceived information and that the perceived information is correct.

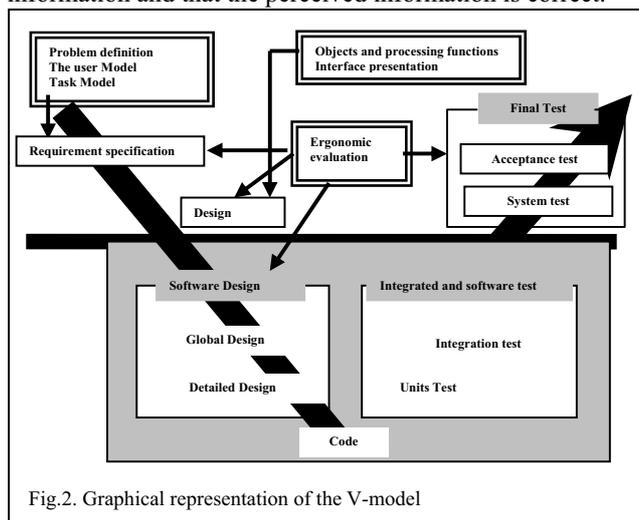


Fig.2. Graphical representation of the V-model

III. SOFTWARE REALIZATION

The design process space is composed of two essential aspects:
 (1) The global design process which establishes the solution to

be implemented and (2) the detailed design process which is oriented toward the software realization.

A The global design process

Our goal is to design a *platform* which locates our system named *SAMAR* (System for Assistance to Maintenance based on Augmented Reality) in its external environment and which describes all the interactions with it. Our analysis of the *platform* design process is based on the UML notation. Our design process area is composed of several interaction situations which allow the technician to explore the whole solutions of design process. To structure our design process area, we consider the input situations (from the user to the system) and the output situations (from the system to the user). The adapters named A_{in} "IN Adapter" are the input devices (a mouse, a microphone,...) which allow to transmit data from the real world to the virtual world. The user can thus interact with the system via these adapters by controlling for example the sequences of the maintenance scenario. The adapters named A_{out} "Out Adapter" indicate output devices (a screen, a Head Mounted Display (HMD), loudspeakers, and a video projector) which allow transmitting data from virtual world to real world. The technician is assisted during his maintenance task. This assistance appears by sending multimedia augmentations through the output adapters A_{out} . The maintenance task is associated to a given breakdown, whose diagnosis is transmitted by an expert to our information processing system *SAMAR* (System for Assistance Maintenance based on Augmented Reality), which display augmentations via output devices A_{out} .

The first entities which appear are: *the User* and *the Machine*. The user handles the machine through tools, which reveals the *Tools* entity. The technician is assisted during his maintenance task. This assistance appears by sending the multimedia augmentation using the output adapters A_{out} . In the case of a collaborative work, the *User* entity will be named *CSBU* (Collaborative System Between Users). It will deal with collaborative work for the same task. The *training* entity will take all the training aspect of the technicians.

The fig.3 illustrates the global diagram of classes.

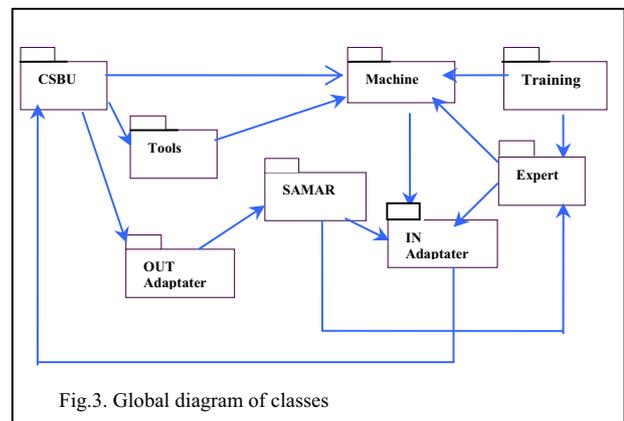


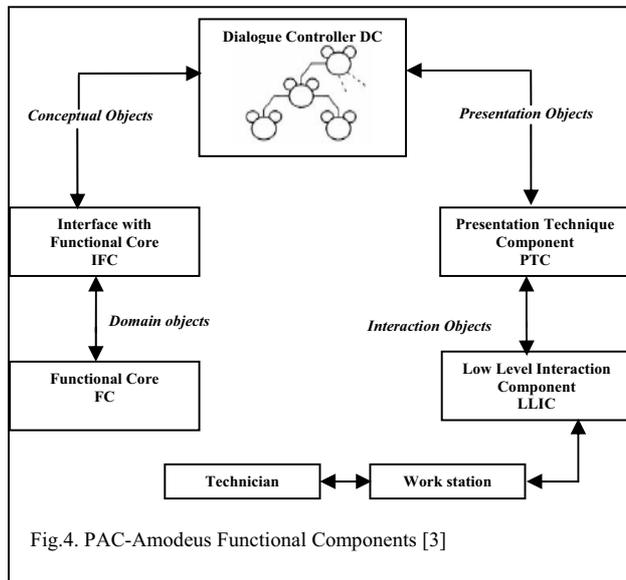
Fig.3. Global diagram of classes

B Software design

The software design of an Augmented Reality system must be based on software model architecture, in order to support its maintenance and its portability in addition to implementing the software properties.

The reference model for an Augmented Reality design systems is hybrid PAC-Amodeus model [12] (see Fig.4).

PAC-Amodeus uses the Arch model as a foundation for the functional partitioning of an interactive system and populates the Dialogue Component with PAC agents: Arch supports the existence of reusable code and defines two adaptors for accommodating style heterogeneity, for anticipating changes and portability. On the other hand, Arch does not provide any guidance about how to structure the Dialogue Component in a way that it is compatible with the user's task requirements. PAC supports task interleaving as well as multiple grains of task decomposition but fails in making the link with existing styles explicit. PAC-Amodeus gathers the best of the two worlds [3].



PAC-Amodeus is a hybrid model that:

- reuses the Arch functional decomposition for its known benefits,
- refines the key component of the architecture, (i.e., the Dialogue Control) in terms of PAC agents to comply with task analysis,
- includes patterns of agents that correspond to recurring situations,
- makes the information explicit flow but leaves the nature of the connectors between the components open,
- does not prescribe any refinement for the components of the arch other than the Dialogue Control, since those may be implemented from reusable code.

The existence of the two real pillars and data processing agrees perfectly with the specificity of the AR systems. The left pillar, *data-processing pillar*, is dedicated to the information processing system, i.e. to the virtual world entities. The right

pillar, *real pillar*, carries the user interaction with the system using the input/output system interfaces. These adaptors are not limited to the mouse and the keyboard, but they include in a general way all the input adaptors (A_{in}) and output adaptors (A_{out}) identified in the *platform*.

Moreover, the user is not the only real entity to be implied in the interaction. The real pillar is thus, responsible as well for the entities integration used in the Augmented Reality system. Consequently, PAC-Amodeus model must be adapted. It must contain several real branches, each one of these branches is thus related to real world entity useful for the interaction (Fig.5). A single Presentation Technique Component (PTC) ensures independence between the concrete implementation of the interaction objects and the Dialogue Controller (DC). Multiple PTC guarantees horizontal independence between the various necessary toolboxes. This independence between PTC of each branch supports the modifiability of the code. Indeed, the use of the locators and other adaptors is independent of the rest of the code. A change of an adaptor is thus simplified and does not require finding the various designers parts of the code to adapt the code to the new adaptor. Reciprocally, the functional core (FC) code is independent of the used adaptors. The addition of branches to the data-processing pillar is inevitable for two principal reasons:

- Technologies variation constituting the Functional Core (FC).
- The re-use of existing FC.

A technologies variation constituting the FC occurs when an application requires the offered functionalities by two distinct FC. For example, an application allowing the results visualization of the specified request by the user is subjected to a database. The FC contains the database as language interrogation, as a relational database. If the application must be able to ask an oriented object database, this one as its interrogation language represents a second connectible FC in the same Interface Functional Core.

To summarize, in PAC-Amodeus model architecture, the task realization out of the user initiative uses the Presentation Technique Component (PTC), which collects the physical actions, then the Dialogue Controller (DC), the Interface Functional Core (IFC) and starts finally data-processing actions at the Functional Core (FC) level. The results then follow the opposite way to the user to perceive feedback information. Some of FC actions can automatically activate the progress of other internal actions in the FC or in the other FC. These tasks, known as automatic, are then invisible to the user and are only used for example, to guarantee the integrity of the data stored in the FC. The Dialogue Controller (DC) is not informed of these actions.

In a symmetrical way, some entities from real world can have an influence on other real entities, without requesting the functionalities offered by the DC.

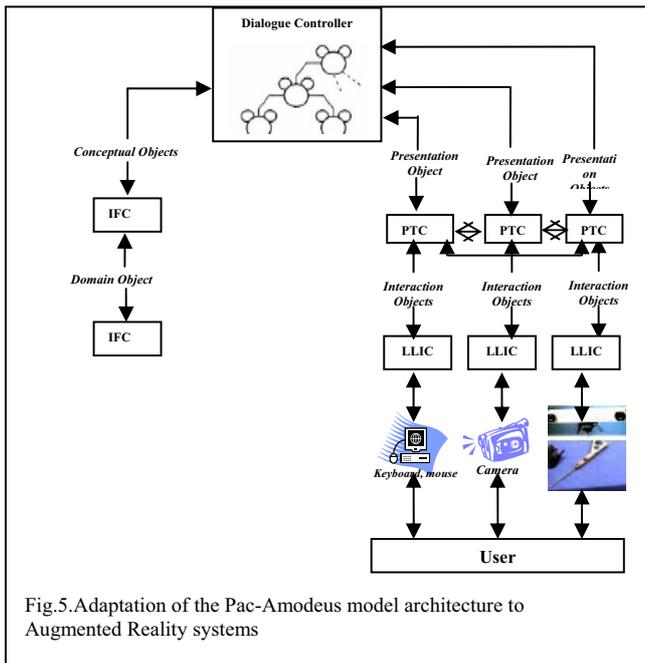


Fig.5. Adaptation of the Pac-Amodeus model architecture to Augmented Reality systems

IV. MODELLING OF OUR SYSTEM USING THE HYBRID PAC-AMODEUS MODEL

In this section, we will describe the three software components composing our system.

A The real pillar

The real pillar of the PAC-Amodeus model architecture is composed of five branches (Fig.6).

The first branch is dedicated to the technician interaction with the system interface. It's related to the graphical interface to allow displaying various augmentations by using graphical library which contains the LLIC and the PTC. The second branch is necessary to answer the technician requests through input interface.

Let us note that these two first branches are dedicated to the technician interaction with the system interface.

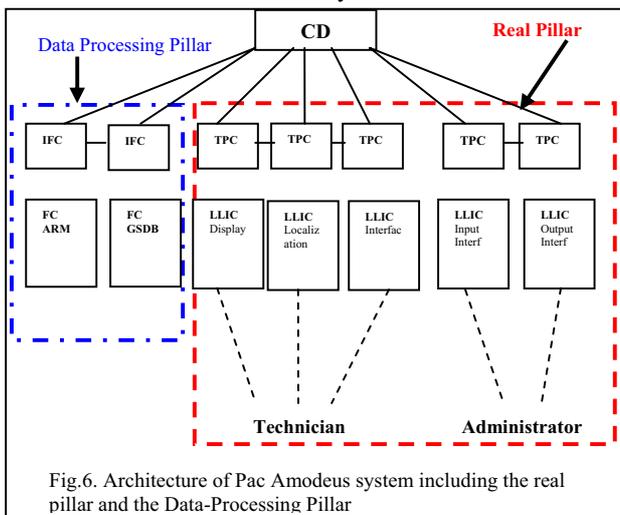


Fig.6. Architecture of Pac Amodeus system including the real pillar and the Data-Processing Pillar

The third branch allows managing the locator data which transmits the technician coordinates compared to a fixed reference frame in the environment to the display PTC without informing the Dialogue Controller. The LLIC and the PTC corresponding to this branch are all the software elements necessary to the locator operation. The fourth and the fifth branches are dedicated to the administrator interaction with the system.

B The Dialogue Controller (DC)

Based on the system functionalities, four agents appear (Fig.7):

- 1) Agent "repair": Manages the sequence of tasks related to the maintenance scenarios.
- 2) Agent "scenario": Consists of supervising the scenario design as well as the various augmentations.
- 3) Agent "breakdown": Manages all the tasks related to the repair orders the administrator gives as well as information pertaining to the occurred breakdowns.
- 4) Agent "DB": (Data Base): Manages the task concerning the data administration related to the maintenance function.

These agents are regrouped under a cement PAC agent that allows establishing the communication between the agents and the Functional Core FC.

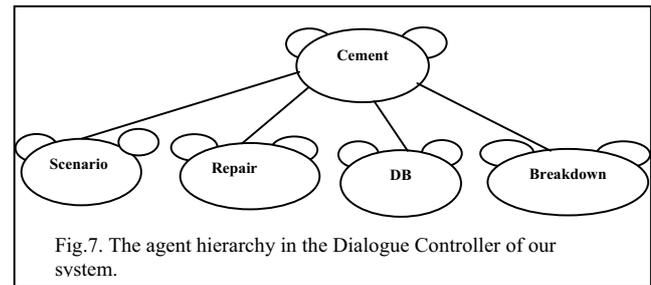


Fig.7. The agent hierarchy in the Dialogue Controller of our system.

C Data processing pillar

The functional part of our system is composed of two distinct functional cores (FC), the FC ARM (Augmented Reality Maintenance) and the FC SDBM (System of Database Management) (Fig.6):

- 1) The FC ARM (Functional Core Augmented Reality Maintenance):

The principal function of the FC ARM is to integrate the virtual entities in the real world. This integration differs according to the augmentation types (sound, video and image). The problem is how to integrate and where to integrate? To integrate augmentations of sound type in the real environment, it is sufficient to have interfaces which allow the sound diffusion like an audio helmet or baffles. For the integration of video sequences augmentation, it is sufficient to have a visualization interface like a screen or videos glasses. For these two types of augmentations, the question where to augment is not asked. It is only enough to position the peripheral so that the technician can listen. As for the video sequences, they cover all the display zone of the peripheral. The problem of the augmentation positioning also does not arise. For the third type of augmentation which is the image, the augmentation

positioning is crucial because the image display must be done at a precise place in the interface display area. If the image covers the entire display surface, the technician will then perceive an image which will hide its real environment. However an image with it only does not have any means. Thus, it should be aligned in the real world. Very significant information is to be exploited in the case of maintenance and the image augmentation is always positioned at the same place compared to the machine because a component has a fixed place. It is consequently enough to establish a machine coordinates and position the augmentation according to these coordinates, which will be stored in a permanent way.

2) The FC DBMS (Functional Core Database Management System)

This Functional Core manages all the data stored in the database. These information are represented in Entities/Associations (E/A) models. Several entities appear:

- 1) The entities “Models machine and machine”. Include the whole information concerning the machine models and all information relating to the machines (machine name, manufacturing ...)
- 2) The entities “Tools”: Contain all information related to the used tools in the maintenance task (tools name, references...)
- 3) The entities “Technician”: contain information concerning the technician (technician name, date of birth...).
- 4) The entities “A_{in}, A_{out}”: include information related to the input an output adapters.
- 5) The entities “Breakdown type”: Contain all the breakdown types of the various machines.
- 6) The entities “Breakdown”: Contain information related to the breakdowns which have occurred in the workshop.
- 7) The entities “Scenario and Augmentation”: Contain information related to the maintenance procedures.

V. TESTS AND EXPERIMENTATIONS

In this section, we will develop the two remaining stages of the V-model, namely coding and tests. The implementation or coding translates the specifications resulting from the design stage detailed into an executable code. The unit test allows verifying whether the modular components of the system answer each one of their specifications. The integration tests are used to check that the modules implemented independently interact correctly. The system test allows verifying that the solution elements expressed in the external specifications file are present. The acceptance tests are used to verify the requirements expressed in the specifications.

A Implémentation or code

The implementation consists in producing an executable code. We give in the following some implementation details of the various modules listed in the detailed design stage.

The passage of E/A model of the database towards the relational model under Oracle is carried out at FC DBMS level.

Let us quote some obtained tables: Environment, Workshop, Model Machine, Machine, tools, Technician, A_{in}, A_{out} Scenario, Augmentation, Scenario_build, Breakdown Type, and Breakdown.

B The Dialogue controller

The Dialogue Controller DC is a PAC agent’s hierarchy. It is, thus, necessary to find the structure which takes the three aspects given bellows:

- Autonomy.
- Heterogeneity of the various received messages.
- Creation / Dynamic Destruction of the agents.

Two key implementations are implemented: The BAL class and the Agent class. BAL class is the key class allowing the communication between the PAC agents. As all the agents are coded in the same manner, we only implemented the cement agent.

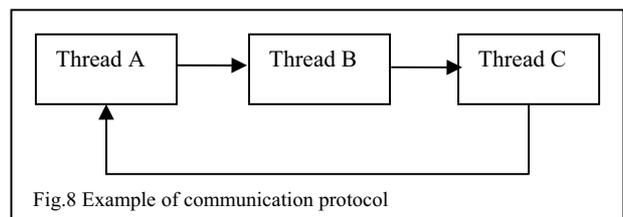
The BAL class is derived from the arbitrary class which will be used as a basis for the message types. It’s named “Generic” and implemented as follows:

```
#ifndef _GENERIQ_
#define _GENERIQ_
using namespace std;
class GENERIQUE
{
public :
virtual ~GENERIQUE();
virtual GENERIQUE* clone()=0;
virtual int recup_codetr()=0;
};
GENERIQUE::~~GENERIQUE()
{
}
#endif
```

C The unit test

The unit test related to two points:

- 1) The asynchronous inter-thread communication: This phase was in particular realized in communicating several messages between three threads in various ways. The fig.8 shows a communication type between three threads.
- 2) The storage at the database level of the multimedia files: This phase is obtained by using only Oracle, by testing the files storage and generation which are done by a Java interpreter integrated into the Oracle environment.



D System test

A maintenance procedure consists in a scenario divided into steps, again divided into actions. An action is a very basic task,

such as to replace a fuse. The creation of scenarios and of the corresponding database of information is very similar to the procedure of creation of classical user paper or electronic documentation.

The tests are carried out on the digital machine named SOMAB400 (see Fig.9), available in the mechanical workshop of CDTA (Advanced Technology Development Centre).



Fig.9. Front view of the SOMAB machine

Fig.10 shows an example of the additional information or augmentation about the fuse (change the fuse FU33).

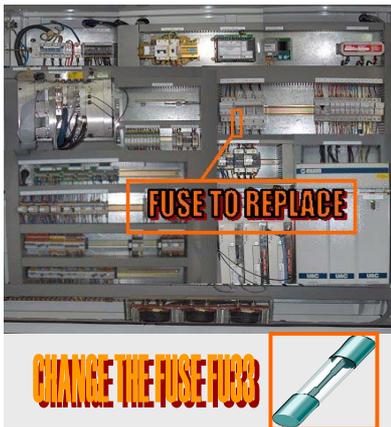


Fig.10. An example of augmentation

VI. CONCLUSION

In order to unceasingly meet the crescent need for the maintenance operators, we proposed in this paper design steps of a prototype for the maintenance assistance based on the Augmented Reality technology which is centred on the interaction analysis between the user, the system and the real world. The design steps and the system realization followed the stages edited by the V-model. Our solution lies in the field of Augmented Reality systems, term which indicates any system combining the real and the virtual worlds

Maintenance takes advantages of the AR technology by giving the possibility to the technician to obtain pertinent information that are not available in his work environment, for this task realization.

VII. REFERENCES

[1] D.Abowd, J.Coutaz and L.Nigay, "Structuring the Space of Interactive Properties", Acte de la conférence EHCI'92, pp 113, 128, 1992.

[2] U. Bockholt, A. Bisler, M. Becker, W. Muller-Wittig, and G. Voss. "Augmented Reality for Enhancement of Endoscopic Interventions". In *Proc. of IEEE Virtual Reality*, pp. 97-101, March 2003.

[3] J.Coutaz, L.Nigay, *Architecture logicielle conceptuelle des systèmes interactifs*, In Kolski C. (Ed.), Analyse et conception de l'IHM. Hermès, Paris, chapitre 7, ISBN 2-7462-0239-5, pp.207-246, 2001.

[4] D.Drascic, "Streoscopic vision and augmented reality".Scientific computing & Automation, pp. 31-34, Juin1993.

[5] L.Da Dalto, "Starmate: using augmented reality for maintenance, training and education", In Virtual Reality International Conference (VRIC), Laval, France, 2002.

[6] A.Dix, A.Finlay, G.Abowd, R.Beale, *Human-Computer Interaction*. 2eme Edition, Prentice Hall, 1998.

[7] C.Gram., G.Cockton, et al., 1996. *Design Principles for Interactive Software*. Chapman & Hall, 248 pages, 1996.

[8] W.E.L.Grimson,, G.J.Ettinger, S.White, P.Gleason, T.Lozano-Pérez, W.M. Wells III, and R. Kikinis, "Evaluating and Validating an Automated Registration System for Enhanced Reality Visualization in Surgery", Proceedings of Computer Vision, Virtual Reality, and Robotics in Medicine '95 (CVRMed '95), Nice, France, pp.3-12, 3-6 April 1995.

[9] J. Kustaborder and R. Sharma "Experimental Evaluation of Augmented Reality for Assembly Training", IWAR'99, San Francisco, October 20-21, 1999.

[10] W.E.Mackay, *Réalité Augmentée: le meilleur des deux mondes. La Recherche*, n° 285, pp. 80-84, mars 1996.

[11] W.E.Mackay, "Augmented Reality: Linking real and virtual worlds - A new paradigm for interacting with computers". In Proceedings of AVI'98, ACM Conference on Advanced Visual Interfaces, l'Aquila, Italy, May 1998. New York: ACM Press, 1998. Available from Internet: <http://www-ihm.lri.fr/~mackay/psfiles/AVI98.AugmentedReality.ps>

[12] L.Nigay, *Conception et modélisation logicielles des systèmes interactifs application aux interfaces multimodales*. Thèse de doctorat en informatique, Grenoble, 1994.

[13] B.Schwald, and all, 2001. "STARMATE: Using Augmented Reality technology for computer guided maintenance of complex mechanical elements". e2001 Conference, Venice-Italy, pp17-19, October 2001.

[14] B.Schwald & B.de Laval, 2003. "Training and Assistance to Maintenance in an augmented Reality environment". Proceedings of the 10th International Conference on Human-Computer Interaction. Volume 3 Cognitive, Social and Ergonomic Aspects Mahwah, New Jersey: Lawrence Erlbaum, pp. 1121-1125, 2003.

[15] B.Schwald, & B.de Laval, "An Augmented Reality System for Training and Assistance to Maintenance in the Industrial Context" Journal of WSCG'2003, pp 425-432, 2003.

[16] Z.Szalavari, E.Eckstein, M.Gervautz, "Collaborative Gaming in Augmented Reality", *Acte du symposium Virtual Reality Software and Technology*, pp. 195-204, 1998.

[17] H.Thomas, W.Piekarski and B.Gunther, "Using augmented reality to visualize architecture designs in an outdoor environment". International Journal of Design Computing Special Issue on Design Computing on the Net (DCNet), Vol.2, 2000.

[18] P.Wellner, W.Mackay, and R.Gold, "Computer Augmented Environments: Back to the Real World", Special Issue of Communications of the ACM, vol. 36, n° 7, pp 24-26, July 1993.