A Reference-Frame for Installing the Student's Computer to Access a Virtual Medicine University

Marilyne Rosselle, Pierre Gillois, Josette Morinet-Lambert, François Kohler

A Reference-Frame for Installing the Student's Computer to Access a Virtual Medicine University

Marilyne Rosselle*, Pierre Gillois**, Josette Morinet-Lambert*** and François Kohler**
*ESIEE-Amiens, 14 quai de la somme, 80000 Amiens
**SPIEAO, Faculté de Médecine, UHP-Nancy I, F54505 Vandœuvre CEDEX
***LORIA-UHP, B. P. 239, F-54506 Vandœuvre CEDEX
*ml.rosselle@ifrance.com, **{gillois, kohler}@spieao.u-nancy.fr, *** morinet@uhp-nancy.fr

Abstract[1]

This research is part of the French-speaking Virtual Medical University (UMVF). French educational medicine-process does not only take place at the university but also in the hospital, at home or in an office. At home, students need to adjust their computer in order to receive and view the educational resources supplied by the virtual university. In the setup-stage, he may need some help, according to his computing qualification level. Therefore we provide him with a setup-system. We specify a reference-frame, which describe pieces of software and hardware that are required in the computer, to be able to work correctly, before the student use it for studying. Moreover, this reference-frame is very useful for a site administrator at the university or in a hospital. We choose UML for modelling the setup-system and the reference-frame. Services concerning the management of the student' computer management (especially confidential issues), its installation and its testing have been enlightened. Those services are not supplied by any well-known WEDS (Web-Based Education Delivery Systems).
Keywords: virtual campus, student's computer, installation, configuration, testing, reference-frame

Résumé

Cette recherche fait partie de l'Université Médicale Virtuelle Francophone (UMVF). La formation française en médecine a lieu non seulement à l'université mais également à l'hôpital, à la maison ou au cabinet médical. À la maison, les étudiants doivent régler leur ordinateur afin qu'il puisse recevoir et restituer les ressources éducatives fournies par l'université virtuelle. Dans l'étape d'installation, il peut avoir besoin de l'aide, selon ses compétences en informatique. Par conséquent nous définissons un système d'installation. Pour ce dernier, nous spécifions un référentiel, qui décrit les logiciels et matériels qui sont requis sur l'ordinateur, pour pouvoir fonctionner correctement, avant que l'étudiant l'utilise pour étudier. De plus, ce référentiel est très utile pour un administrateur de site universitaire ou hospitalier. Nous modélisons le système d'installation et le référentiel en UML. Nous avons souligné des services concernant la gestion de l'ordinateur de l'étudiant (en particulier les questions de confidentialité), son installation et son test. Ces services ne sont assurés par aucun WEDS bien connu.
Mots clés : campus virtuel, installation du poste de l'étudiant, configuration, test, référentiel.

Introduction

This research is part of the French-speaking Virtual Medical University (UMVF). UMVF is a national consortium (international in the long term) including the nine biggest French medicine universities, in partnership with eight qualified firms of e-learning, video, transmission and edition. This very ambitious university for medical education is based on "Web-Based Education Delivery Systems" or WEDS (Bourguin and Derycke 2000) (Doube 1999) (Heift and Nicholson 2000) (Peylo 2000). The medicine-educational process does not only take place at the university, but also in the hospital, at home or in an office. This diversity of access places is very important to allow the long-life education, substantial in medicine.

In this paper, we point out that many services are not supplied by any well-known WEDS (Barker and Pilkington 2000) (Beuschel et al. 1999) (Hazari 1998 URL) (Lowery 1999) (Midoro 1999) (Virvou and Moundridou 2000). A student who has just bought a "standard" computer may not easily access to the virtual university. Since this student's computer will have to retrieve resources from the WEDS and restore them, the student at home must adjust its environment. he should set-up and configure his computer, even if he is not computer-literate. According to his computing qualification level, he may need a setup-program. Therefore, we start from the student's needs related to the management of his computer to use services provided by a specific virtual university. This requires modelling the particular conditions in which a student uses its computer. Moreover, in order to guarantee that the computer is ready to work with the virtual university, we

[1] This publication relates to a research which took place within the framework of a Spi-eao/UMVF contract

define a "reference-frame", which records all software and hardware to be sought on the student's computer, according to the pedagogical resources available on the virtual university. For example, for a video resource (type of document), a video player (software) and a video card (hardware) are required.

First we present UMVF and the medicine context. Second we establish who is the user that have to set-up the computer. Third we describe a part of the UML-formalised setup-model. And fourth, we show the reference-frame.

UMVF and Medicine Context

At the time of its constitution in 1999, the consortium defined the requirements of a virtual campus (Le Beux et al. 2000)for building, validating and experimenting methods and tools for medical education. The consortium made an inventory of available resources (pedagogical support and tools) (Slotte, Wangel, and Lonka 2001) (Larsen et al. 2001) (Harris, Leaven, Heidger, Kreiter, Duncan, and Dick 2001). It also defined new tools and services, which were not yet developed in WEDS, for example, research engines, which are specific to medicine domain (e.g. cardiovascular) and which use source validation (Evidence Based Medicine).

The virtual campus define by UMVF (CVU - The Virtual Campus of UMVF), should fit at all the parts of educational system i.e. administration, organisation, delivery and evaluation of knowledge. It should be platform-independent i.e. teacher and learner can choose any type of machine (PC, Mac …) and operating system (MacOS, Windows, Linux ….). It should be WEDS-independent i.e. each university is free to use the WEDS of its choice (Campus Virtuel, WebCT, Learning Space).

Even if CVU is specifying as WEDS-independent, the consortium UMVF has chosen a particular WEDS as a feasibility demonstrator, according to the ORAVEP study (ORAVEP URL). This WEDS, CVA ("Campus Virtuel®" of Archimed i.e. Virtual Campus) is published by the "Archimed" firm (CVA URL), on the basis of a research-prototype developed at the university of Lille 1, came out from the European project CO-LEARN. It can be evaluated on the CVA' site.

French educational medicine system is shifting from a local evaluation (local to each medicine school) to a national evaluation (competitive entries in specialities). Therefore that could allow students to share pedagogical resources and to access inter-university courses from any place.

UMVF is based on secured-network, accessible by the users as well on the University Campuses as in the University Hospital Complexes (CHU) – intra-hospital - or as on any private computer. In the case of an access from a hospital, the cohabitation with the health system generates rights and confidentiality problems, generally not anticipated in the WEDS.

Who is the User, that Set-up the Computer?

The less a user is computer-literate, the more a setup-program is required. Hence we have to consider the user's computing level and his localisation.

Defining the profile of this user

From the characteristics of the medical education, we can deduce that the dialog WEDS-user should consider additional information in the usual user's profile.

First we consider various computing levels. The aim is to adapt vocabulary and functions. The possible values are expert and beginner. To the "beginner" value corresponds a dialogue for which the used vocabulary should be the simplest. The user-actions should also be as effortless as possible. The ideal case is: the beginner student launches the setup-program. This is the only thing he has to do before access CVU and use it optimally. To the "expert" value corresponds a dialogue for which a technical level and some technical vocabulary are necessary. This type of user should check the modifications, which are made to its usual configuration. He should also decide if he accepts these modifications or not.

Secondly we consider various places where the computer is setup. Indeed with each place its constraints. The student at home could adjust its computer however he wants. But the student at the university or in a hospital cannot adjust it. A site administor applies rules and procedures imposed by his institution university concerning installation and configuration of computers (cf. safety requirements, limitation of Internet access, etc.). To consider these constraints, the administrator of site has to make. Therefore the set-up program should let him decide.

In addition, the network characteristics (flow, cost, etc.) in each place should be known: high cost at home paid by the student / fixed price paid by the institution; low-flow at home / high-flow at the institution, etc. This allows deciding the types of resources, which can be broadcast. Furthermore, on the site of the hospital, it is necessary to consider the constraints of confidentiality of information circulating on the network and the need for inter-connecting the hospital network with the university network.

Therefore, the possible values to describe the place of the computer are home, hospital and university.

And third we consider the various types of computers, on which CVU is access. The student may have a private computer at home (or at his office). But, he also uses either an individual or a public computer (at the university or in a hospital - in multimedia room, self-service room, etc.) or a shared computer (in an office). In this case, he should get its usual environment. Privacy of the data he consults must be guarantee. This implies that buffers (cookies, etc.) are deleted when he disconnects from CVU. Thus, the possible values to describe type of computers are private, individual and public.

Because the student is itinerant, we need to define conditions of Education on different types of computers, set-up by users with different computing levels, located in different places.

Extension of the user's types in WEDS design

WEDS design usually considers the following types of users: student, administrator of the WEDS, teacher, author, administrator of the formation. These users are identify in WEDS modelling, during the standard-use stage of the WEDS's life. But in the installation stage, we need to define new users.

Usually, WEDS publisher provide a few lines to describe the computer's requirements. We experiment that few lines are not sufficient to describe a computer set-up. To model these set-up requirements, we define a new role assumed by the "setup-user". So we extend the set of the WEDS' users with the setup-user. This setup-user could be a student at home or an administrator of site (at the university or in a hospital).

After being set-up, the computer should be tested. Indeed, the setup-user should verify that everything is all right before allowing a student to use the computer. This test may take place at any moment by another user. Therefore we define a "tester". A sample-lesson is then required. A sample-lesson provide a core of test, representative of what can be done with the WEDS. For example, this sample-lesson should possess various types of resources (sound, video, text, etc.) and imply various types of tools (research engine, mail, chat, forum, etc.). With this sample-lesson, the tester should be able to check that each resource can be restored (sound eared, video seen and eared, text seen, etc.). He also should check, he could use all provided tools (search a particular document with the research engine, ask a question to the teacher by mail, collaborate with other student by chat or by using the forum, etc.). This check-up is useful to determine if a plug-in, a piece of software or a sound device is missing in the computer. As for the setup-user, the less the tester is computer-literate, the more an integrated test-program is helpful.

In the installation stage, the users of the student's computer are the student, the tester and the setup-user. The roles of setup-user and of tester are essential to guarantee the quality of the delivered services. These types of users add a dimension to the problem of delivering education on the Web. We decide to formalise the setup-user's profile and the setup-program.

Results: Formal Specifications

Description of the used formalism

To formalise the specifications of the setup-system, we chose a graphic modelling in UML. This Section describes the elements of UML model we use in this article.

Definitions

Modelling in UML lies on a model of use-cases. This model is elaborate starting from the user's needs. The crucial role of model of use-cases is the expression of the functions of the system and its behaviour. A use-case model describes the behaviour of the system (here the setup-system) to model in term of:

> Cette recherche fait partie de l'Université Médicale Virtuelle Francophone (UMVF). La formation française en médecine a lieu non seulement à l'université mais également à l'hôpital, à la maison ou au cabinet médical. À la maison, les étudiants doivent régler leur ordinateur afin de recevoir et regarder les ressources éducatives fournies par l'université virtuelle. Dans l'étape d'installation, il peut avoir besoin de l'aide, selon son niveau de qualification en informatique. Par conséquent nous définissons un système d'installation. Nous définissons un référentiel, qui décrit les logiciels et matériels qui sont requis sur l'ordinateur, pour pouvoir fonctionner correctement, avant que l'étudiant l'utilise pour étudier. De plus, ce référentiel est très utile pour un administrateur de site universitaire ou hospitalier. Nous modélisons le système d'installation et le référentiel en UML. Nous avons souligné des services concernant la gestion de l'ordinateur de l'étudiant (en particulier les questions de confidentialité), son installation et son test. Ces services ne sont assurés par aucun WEDS bien connu.

environment: actors;

* functions expected from the system: use-cases;

* and relations between the use-cases and the actors: use-case diagram.

Actors are not part of the system. They represent any person or entity likely to interact with it. They can give information to it and/or receive information from it.

Use-cases describe the functions of the system. They model a relation between an actor and the system, in other words, a service provided to the actor by the system (relation labelled with "communicate").

A use-case diagram is a graphic view of whole or part of a system: its actors, its use-cases and their relations. In such a diagram, " Extend " labels an association between a use-case and an optional use-case, which indicates an optional behaviour.

These UML elements describe the "components" of the system.

Main actors of the setup-system

The "SetupUser" is the user, who launches the installation itself with the setup-system. He is an UML actor.

The setup-program depends on the UMVF virtual campus (CVU). Indeed, the setup-program has to check, on the student's computer, the existence and the configuration of the required hardware and software. "What is to check" is variable according to a given version of CVU (depending of a chosen WEDS). To consider that, the set-up interacts with an actor called "CVUReference" (reference-frame for CVU). CVUReference supplied the setup-up systems with the pieces information record in the reference-frame. We look further into the concept of reference-frame in a next section.

The system requires inspecting the student's computer in order to check hardware and software and their

configurations as defined by CVUReference. The student's computer holds information necessary to the setup-system. We thus define an actor called "StudentComputer".

Main use-cases of the setup-system

From the actors' definition , we can draw the main functions expected from the system:

* the system provide the setup-user the faculty to set up the student's computer;

* the system interacts with StudentComputer to consult the information held by this last and to detect the required installations or configurations of hardware and software;

* the system interacts with CVUReference to seek of its elements on the student's computer.

From each function, we extract the use-cases in term of objectives of the system justified by at least one actor's need.

We define the use-case "ComputerSetup" to correspond to the first function.

The second function is only used by the first one. Moreover, the StudentComputer actor has no particular need with respect the system. Therefore, a use-case definition is not required. However, after the realisation of the previous use-case (ComputerSetup), the setup-user may want to consult information about his computer. These information are, for example, information collected on its computer, proposals of installation and configuration, modifications made or to bring, warnings consecutive to the made-choices or the specified-needs (potential problems), special events occurred during the realisation of the use-case "Consult", etc. The use-case "ConsultInformation" provides the setup-user with this ability.

In the same way, the third function is used only within the framework of the first. Moreover the CVUReference actor has not particular need with respect to system. Therefore, a use-case definition is not essential. However, the setup-user may want to consult the reference-frame. Therefore we define a specific use-case to provide him with this ability. This use-case is called "CVUReferenceConsult".

Main use-case diagram of the setup-system

Previous sections are both abstract on Figure 1.

On this figure, a character represents an actor. The human one is on the left side, the entities are on right sides. An arrow represents dialogs between an actor and a use-case. The consultation of information about the computer is not absolutely required in a setup-program. So is the consultation of the reference-frame.

The setup-user's context cover different realities, which are detailed with in the next section.

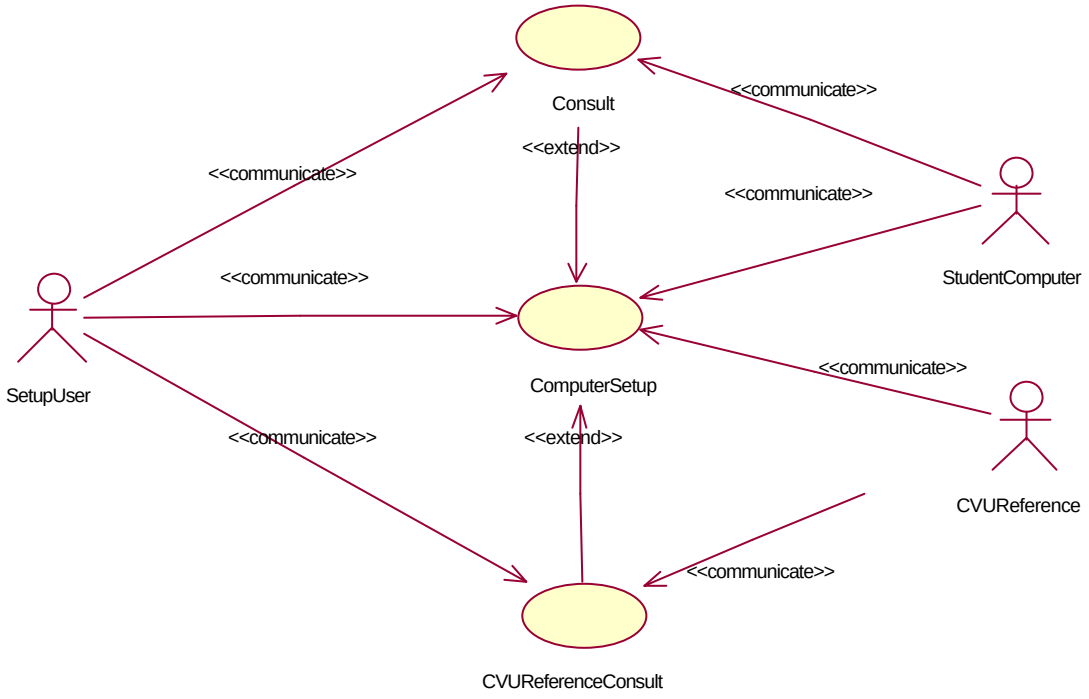

Figure 1 – Use-case diagram for setup-system

Specific needs of the setup-user

A "SetupUser" is an actor, who launches the set-up on the student's computer. He has three attributes:

* “place” describe the place of the computer, from where the setup-user launches the setup-program. Its possible values are home, hospital and university.

* “type” describe the type of the computer, from where the setup-user launches the setup-program. Its possible values are private, public and individual.

* “computingLevel” describe the computing level of setup-user. Its possible values are expert and beginner.

Set with these three attributes, the setup-user’s actor is sufficient to adapt the behaviour of the setup-system.

In this Section, we have presented parts of the setup-system’s UML-model. This system uses a reference-frame to assess the computer. Parts of the model of the reference-frame are presented in the next section.

Reference-Frame

The reference-frame describes hardware and software that should be present on the student’s computer. Moreover, it describes the required configuration of these software and hardware. The aim of this information is to guarantee that the computer is able to function correctly with a given version of CVU. The formal reference-frame is called “CVUReference”.

An usual example of a WEDS requirements for the student’s computer look like this: “your computer must meet or exceed the following system requirements: Internet Explorer 5.01 Service Pack 2 or Netscape, Pentium processor recommend, 32 (MB) megabytes (for Windows 95,Windows 98 or Windows Me), 64 MB (for Windows NT 4.0 with Service Pack 3 or higher) or 128 MB (for Windows 2000) of random access memory (RAM) minimum, Mouse, Keyboard, Modem/Network connection.”

These basic requirements are not sufficient to guarantee that all types of documents send to the student will be restored correctly. For example, a medicine lesson about the heart-breath requires audio record. Consequently, the computer needs to have some speakers, an audio card (hardware) with its drivers and an audio player (software). Moreover, the navigator should link the audio type of document to a plug-in or an external application. If one of these elements is missing, the student’s computer is not able to get this kind of lessons. All these elements should be found in the reference-frame.

To define the reference-frame, we first we identify the physical elements (hardware and software), which it should be composed with. Then, we point out some questions that show physical elements are not sufficient.

Hardware and software elements

We start from the types of documents supplied by the WEDS. These types of documents are limited because UMVF consortium has selected some of them, on the bases of existing standards. Consequently, the student’s computer should be able to restore these types of documents.

Then for each type of document, we get back the type of software to restore the document (audio player, video player, etc.). We find out some example of application or plug-in of each type, for each main Operating System (OS selected by the consortium).

After doing this, we find out hardware (and drivers) requires by each type of document. Hardware often depends on the OS. We also determine other hardware and OS pieces of information, which are depending on software (RAM, hard-disk-space required during the software install and after its install, etc.).

Then, we detect the specific configurations of each piece of software and hardware.

Lastly, we can check that all software and hardware is available for the selected OS.

Other elements

Several questions appear when software, hardware or their configuration is not exactly what is required. What should the setup-program do or propose to? To illustrate a part of the problem, we describe the software problems with a case study. Suppose:

* the setup-program is launch by a beginner student (thus we can admit that the setup-program take all decisions);

* the student’s computer is a personal computer with Windows 2000 OS correctly installed and with an internet access;

* the student’s computer has a sound card with its drivers well installed (that hypotheses is for concentrate on software problems only);

* the WEDS “send” a lesson to the student with a mpeg (Mpeg-1 Audio Layer 3) audio resource (extension “.mpeg” and mime type “audio/mpeg”);

* the reference-frame associates mpeg audio resources with a set of some existing applications (RealPlayer, Winamp and FreeAmp) and some plug-in (Quicktime 5, Music Player and Media Player) for PC/Windows 2000.

Several cases can occur.

Case 1: the student’s computer uses one of the pieces of software associated with mpeg audio resources in the reference-frame. The mime type of this kind of document is linked with the navigator. The piece of software is a sufficient version. Therefore the student can ear the mpeg audio resource. It’s the ideal case.

Case 2: the student’s computer has no piece of software associated with mpeg audio resources. He cannot play an mpeg sound resource. The setup-program should choose a piece of software in the reference-frame set.

But on which criterion should it make this choice (cost, place required on the hard-drive, number of other types of document cover by this piece of software, etc.)? Should the setup-program privilege a plug-in instead of an application? Moreover the chosen piece of software should be compatible with all characteristics of the computer: name and version of the navigator, characteristic

of the OS, free space in memory (RAM) and on the hard disk, processor, etc.

Case 3: this case is the same as Case 1, but the piece of software is an obsolete or insufficient version. Should the setup-program install the new version of this piece of software? Or should it install the "best-choice" according to the criterions presented in Case 2?

Case 4: the student's computer uses a piece of software unknown in the reference-frame. Let us call it "Esoteric". Esoteric is associated with mpeg audio resources in the navigator. Consequently the student could ear the mpeg audio resource. However what should the setup-program do? Which strategy adopt? Should setup-program keep the "Esoteric" piece of software or to install a piece of software in the reference-frame set? In the case where a piece of software of the reference-frame set is chosen, the problem of choice is still the same than in case n°2.

This case study is a simple one. Indeed a beginner student does not have to take all decisions. But if the setup-user is a site administrator, then he could accept or not each install-proposition made from the setup-program. Therefore, what happens if the setup-user refuses the installation of piece of software chosen by the setup-program? Could the computer function correctly? What is "blocking"?

All these questions should obtain answer, depending on each chosen WEDS requirements and on consortium decisions. These answers should appear in some manner in the reference-frame.

In the following Section, we illustrate a model of the reference-frame, including constraints introduce in this Section.

Part of the model of the reference-frame

The reference-frame is composed of some sets, modelled with lists. We propose to describe a kind of files with his mime type (instead of with his extension). All (and only) the aspects evoked previously are shown in Figure 2.

In Figure 2, CVUReference posses four lists. Each types of files is linked with a list of pieces of software, sorted relatively to the criterion and strategy defined. Each type of files is also linked with the required Hardware (by the means of a piece of software). This figure is a UML class diagram. A link between two classes indicates an association with its cardinality. For example, one mime type (MimeType) is associated to one sorted list of pieces of software (SoftwareSortedList). A rhombus on a link indicates an aggregate. For example, one sorted list of pieces of hardware (HardwareSortedList) is composed by at least one piece of hardware (HardwarePiece). Arrow indicates a generalisation relation.

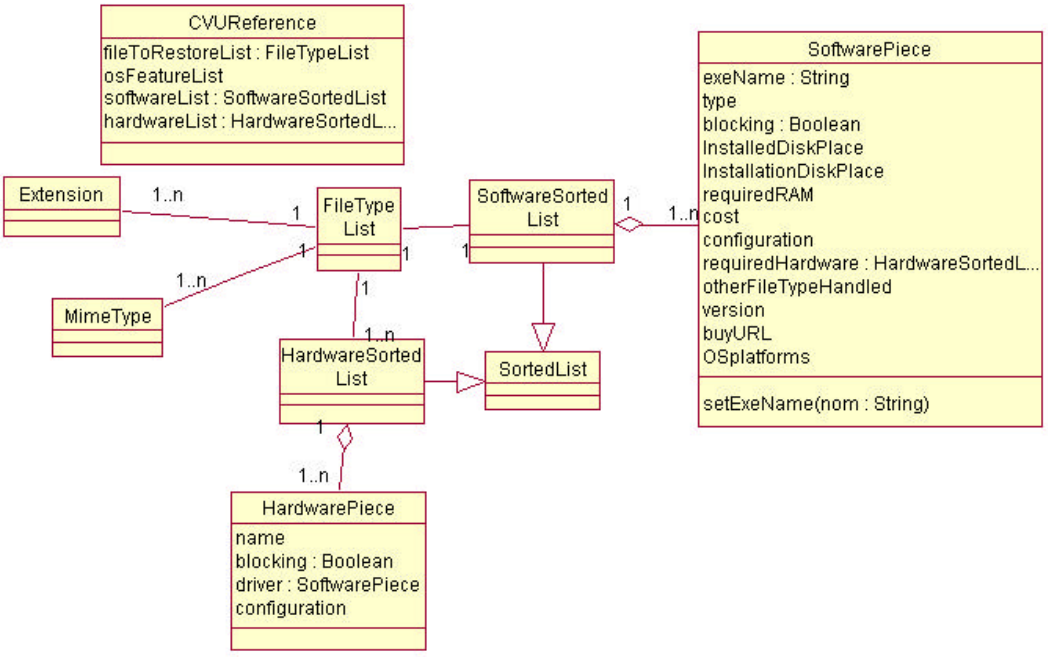

Figure 2 – Part of the logic model (UML) of the reference-frame

New needs linked with the reference-frame

WEDS should provide services to manage the reference-frame. WEDS managers should decide:

* where the reference-frame is located? (On a site?)
* who validate the reference-frame? (It supposes to define a new type of users).
* who update the reference-frame? (It suppose to define services to update it and to define new type of users)

* in which form and by which access-mode is the reference-frame available? (A database to invoke or a structured text file to download?)

The answers to this question during the design of a WEDS implies the definition of new kind of users and new services.

Conclusion and Trends

In the research presented here, we focus on the student's needs, especially concerning the setup-stage of the student's computer. Our results are the four following ones. First of all services concerning the management of the student computer installation and its testing have been enlightened. Those services are not supplied by any well-known WEDS. Other services are also required because of the specific medicine context (e.g. emptying caches before quitting public computers in universities) and because of the possible home uses (e.g. calculating the connection costs and download times). Secondly, to model the particular contexts of where the student uses the computer, we have extended the student's profile with his computing level and his localisation (place and type of computer). Third, we list several categories of documents the computer should restore in order to use the resources available on the WEDS. Then, from these categories, we define the required types of software and hardware. Finally, all this information has been recorded in a reference-frame, which is compared to the features of the computer. All these requirements have been clearly specified in UML and can now be implemented. And fourth, WEDS design usually models the standard-use stage. We have defined new categories of users WEDS design should include in: the setup-stage: the setup-users, the testers and the reference-frame's managers.

Three parts of these results could be generalised. First, the reference-frame could be generalised to other kinds of setup-programs and update programs. Secondly, the extension of the setup-user's profile could be valuable for other categories of users. For example, the teacher's localisation should be considered. And third, the method we have used for the setup-user's computer could be applied to other users of WEDS (teachers, authors, etc.) and extended to users of any distributed system.

The trends of this work are the fourth following. First, the student's needs are overall already well considered in WEDS's design during the standard-use stage. We have specified services for the installation stage. In future work, we will focus on other particular uses of the WEDS. For example, a use at home implies constraints that should be modelled. This could generate new services, for example to be able to handle confidentiality or to allow the adaptation of the interface for a disable person. Indeed, what happens in most cases when a user enlarges the font size in his navigator? Secondly, we have underlined the tester's function. In future research, we will formalise the tester's services and profiles. We acutely insist on the need of a sample-lesson (provided by the WEDS manager), in order to prove whether the computer is able to work correctly or not, before the student use it for studying. Third, pedagogical resources included in UMVF are already described using normalisation (Dublin Core and IEEE P1484). These standardised descriptions should be extended to integrate new pieces of information, for example, validation, mime-type, download size, **type of restoration software and version of software**. Description of software that restores these pedagogical resources should also be normalised. And Fourth, education in a hospital implies to consider the connection between the university network and the hospital network. It raises some additional problems, especially if the student is requested to consult existing clinical cases. The use of communication standards of medical image and information like DICOM or HL7 should also be envisaged.

This research domain is still raising issues, which could become valuable supports to improve WEDS in the years to come.

References

(Barker and Pilkington 2000) Barker T. and Pilkington R., Improving the Effectiveness of World Wide Web Delivered Courses. 5th International Conference, Intelligent Tutoring Systems, ITS'2000. Montréal, Canada, June 2000. Young Researchers Track Proceedings, P61-64.

(Beuschel et al. 1999) Beuschel W., Bork A., Hughes C., McMahon T. G., Serdiukov P., and Stacey E., Better Learning Online? IFIP TC3 WG3.2/3.6 International Working Conference on Building University Electronic Educational Environments. Irvine, California, August 1999. Franklin S. D. and Strenski E. Eds., p233-251, Kluwer Academic Publisher.

(CVA URL) servuvmf.archimed.fr/ABV/

(Bourguin and Derycke 2000) Bourguin G. and Derycke A., A Reflective CSCL Environment with Foundations Based on Activity Theory. 5th International Conference, Intelligent Tutoring Systems, ITS'2000. Montréal, Canada, June 2000. Gauthier G., Frasson C. and VanLehn K. LNCS 1839, p272-281, Springer.

(Doube 1999) Doube W., Commonalties and differences in Electronic Educational Environments. IFIP TC3 WG3.2/3.6 International Working Conference on Building University Electronic Educational Environments. Irvine, California, August 1999. Franklin S. D. and Strenski E. Eds., P253-270, Kluwer Academic Publisher.

(Hazari 1998 URL) Hazari, S. I. Evaluation and Selection of Web Course Management Tools. sunil.umd.edu/webct/

(Harris, Leaven, Heidger, Kreiter, Duncan, and Dick 2001) Harris T., Leaven T., Heidger P., Kreiter C., Duncan J. and Dick F., Comparison of a virtual microscope laboratory to a regular microscope laboratory for teaching histology. Anat Rec. 2001 Feb;265(1):10-4.

(Heift and Nicholson 2000) Heift T. and Nicholson D., Theoretical and Practical Considerations for Web-based Intelligent Language Tutoring Systems. 5th International Conference, Intelligent Tutoring Systems, ITS'2000. Gauthier G., Frasson C. and VanLehn K. Montréal, Canada, June 2000. LNCS 1839, 354-362, Springer.

(Larsen et al. 2001) Larsen O. V., Haase J., Ostergaard L. R., Hansen K. V. and Nielsen H., The Virtual Brain Project--development of a neurosurgical simulator. Stud Health Technol Inform. 2001;81:256-62.

(Le Beux et al. 2000) Le Beux P., Le Duff F., Fresnel A., Berland Y., Beuscart R., Burgun A., Brunetaud J. M. , Chatellier G., Darmoni S., Duvauferrier R., Fieschi M., Gillois P., Guille F., Kohler F., Pagonis D., Pouliquen B., Soula G. and Weber J., The French Virtual Medical University. Stud Health Technol Inform. 2000;77:554-62.

(Lowery 1999) Lowery V. Moving towards an effective Electronic Training Environment. IFIP TC3 WG3.2/3.6. International Working Conference on Building University Electronic Educational Environments. Irvine, California, August 1999. Franklin S. D. and Strenski E. Eds., P187-199, Kluwer Academic Publisher.

(Midoro 1999) Midoro V. Modelling Online Education. IFIP WG 3.1 and 3.5 (in co-operation with 3.6) Open Conference. Communications and Networking in Education : Learning in a Networked Society. Aulanko-Hämeenlinna, Finland, June 1999. P239-246, Yliopisto (Helsinki).

(ORAVEP URL)
www.oravep.asso.fr/synergie/pourtous/kiosque/k_publ.htm

(Peylo 2000) Peylo C., W2-Adaptative and Intelligent Web-based Education Systems. Workshop in 5th International Conference, Intelligent Tutoring Systems, ITS'2000. Montréal, Canada, June 2000. Gauthier G., Frasson C. and VanLehn K. Eds. LNCS 1839, p663, Springer.

(Slotte, Wangel, and Lonka 2001) Slotte V., Wangel M. and Lonka K. Information technology in medical education: a nationwide project on the opportunities of the new technology. Med Educ. 2001 Oct;35(10):990-5.

(Virvou and Moundridou 2000) Virvou M. and Moundridou M., Modelling the Instructor in a Web-based Authoring Tool for Algebra-Related ITSs. 5th International Conference, Intelligent Tutoring Systems, ITS'2000. Montréal, Canada, June 2000. Gauthier G., Frasson C. and VanLehn K. LNCS 1839, p635-644, Springer.