Acta Universitaria
DIRECCIÓN DE APOYO A LA INVESTIGACIÓN Y AL POSGRADO

Formal Definition of Collaborative Spaces.

Sergio Arzola-Herrera*, Josefina Guerrero-García*, Juan Manuel González-Calleros*, Claudia Zepeda-Cortés *

ABSTRACT

Collaborative spaces are widely used for diverse organizations and purposes. Despite the fact that technological solutions exist there is a lack of methodological support to develop such environments. In this paper we illustrate how FlowiXML methodology can be used to develop collaborative spaces using a real life case study. The benefits of the resulting system are evaluated and the results are discussed.

RESUMEN

Los espacios de colaboración son ampliamente usados por diversos organismos y propósitos. A pesar de que existen soluciones tecnológicas, ay una carencia de soporte metodológico para la elaboración de estos entornos. En este artículo se ilustra cómo la metodología FlowiXML puede ser usada para desarrollar espacios de colaboración con un caso de estudio real. Los beneficios del sistema resultante es evaluado y los resultados son discutidos.

Recibido: 10 de Enero de 2012
Aceptado: 14 de Febrero de 2012

INTRODUCTION

Systems are developed in order to facilitate the user tasks; however a lot of systems do not do it very well, because they are complex and not accord to the user needs. This happens because usually programmers and designers take too much importance about the system developing, such as how to implement the information into the database or the connection with servers, and take the user in second place; even some developers do not consider the user, for whom is destined the system. We can see this by reviewing the actual system interfaces which does not accord with the user needs. Here we present a methodology which is focused in the processes and user's tasks into a workflow. This methodology pretends to save time and bring a system with quality to the user according to his needs. Also we present a case study of this methodology. This paper is structured as follows: In Section *Realted Work* we present a background of FlowiXML Methodology. In Section *Extending and Enriching Heuristic Evaluation* we present our case study which is about a divulgation platform for scientific research, in this section we make the task identification, the process modeling and the task modeling for this example. In Section *Results and Discussion* we present the evaluation. Finally in Section *Conclusions* we bring the conclusions and future work.

BACKGROUND. FLOWIXML METHODOLOGY

FlowiXML [5][7][8] is a method that provides means to formally design a Workflow Information System (WfIS). Workflows are activities involving the coordinated execution of multiple tasks performed by different resources to achieve a common business goal. A task defines some work to be done by a person, by a software system or by both of them. Specification of a workflow involves describing those aspects of its component tasks (and the resources that execute them) that are relevant to control and coordinate their execution, as well as the relations between the tasks themselves.

Information in a workflow mainly concerns when a certain task has to start, the application information needed for performing the tasks, the criteria for assigning the task to resources, and the ending of the task. The development of workflow technology can be traced back to various origins, such as: office information systems [1], computer supported cooperative work (CSCW), imaging and document management as well as advanced database technologies, all relevant to e-learning systems. A Workflow Management System (WFMS) allows

Palabras clave:
Componente; espacios colaborativos; interfaz de usuario; desarrollo basado en modelo.
Keywords:
Component; collaborative spaces; user interface; model based development.

* Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, 14 Sur y Av. San Claudio, Edificio 136, Ciudad Universitaria (CU), c.p 72570 , Puebla, Mexico.E-mail: satyendraser@gmail.com, jguerrero@cs.buap.mx, juan.gonzalez@cs.buap.mx, czepedac@gmail.com

both to specify workflows and to control their execution. During a workflow execution, a WFMS has to schedule tasks (including their assignment to resources) on the basis of the (static) workflow specifications, of the (dynamic) sequence of events signaling the completion of tasks, of available data, and of generic events produced within the execution environment. FlowiXML assists designers in specifying a Workflow Information system (WfIS) and some guidance on how to derive its corresponding user interfaces. It is composed on the following major steps:

- Workflow information system requirements. This is the result of the elicitation of the organization. We assume that there are means such as: interviews, direct observation, to collect information that will serve as input to identify workflow element. This step corresponds to the requirements of the problem.

- Workflow information system design. This step includes modeling of: workflow, organizational units, jobs, user stereotypes, processes, workflow allocation patterns and tasks. Mapping the workflow specification into a workflow information system.

- Workflow information system development. We consider the development of UI for: task models, allocation patterns, agendas, worklist. Even that it is not explicitly defined we considered that the implementation of a workflow manager is possible based on the workflow designed in the previous step.

In figure1, forward and backward arrows denote the propagation of information from one model to another. For instance, a new task model must make available a task for a process model and vice versa, a new task in a process model might be detailed with a task model. Jobs, user stereotypes and organizational modeling just affect the workflow model. Then the workflow model makes them available for process modeling and task modeling. This particular aspect of concepts propagation was significantly useful for the software tools that support our methodology. The system design is an activity that can start from any model except for the task allocation (dash lines model) because it needs tasks and resources already defined. The design of a workflow permits designers to identify concepts freely and to start to detail based on their preferences. One designer must prefer to get into details of task modeling before describing a process model. Once the task models are ready then it can model the processes that then are arranged to represent the workflow. Another designer might have a better understanding of the problem with the workflow model (more abstract view of the problem) and then start to refine by adding process models and finished with task models. There is no constraint on the starting and end point for modeling just to be sure keep the traceability of the concepts that are shared in different models (task model is part of process and a process model is part of a workflow model).

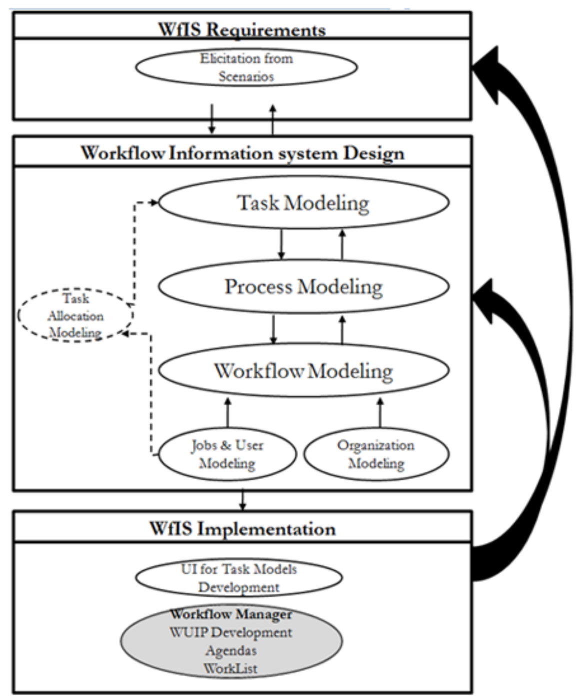

Figure 1 . An overview of the User Interface Development Method

DIVULGATION PLATFORM FOR SCIENTIFIC RESEARCH

Here we present a case study for a platform which pretends to solve a common problem in scientific research, which it is divulgation.

Often published articles are not correctly divulged, because they are only known by people that is around the researcher or by people that is involved in the area. Commonly scientific groups publish their works in a HTML page; the problem is that most of its pages are not frequently updated.

Also another main problem are the conferences and workshops, because if they are not very known, they are just known by the host university, and they are only spread by an e-mail list, which is integrated by people who has participated before.

This is also a big problem for students, because if they want to know about what is done in a specific area, and if the student does not know the researchers or the conferences about that area, then it would be very difficult to the student to find the information.

The way that we propose to solve these problems is through a platform which integrates administration for the content, in a social way. For instance, if a researcher A collaborates with researcher B in an article for the project X, which is presented in the P Conference, and if researcher A upload the work, then the platform will link the published content into the personal page of researcher B, in the page of the project X and in the page of the conference P.

Task Identification

This step is concerned with understanding the problem by studying an existing organizational setting; the emphasis is put on identifying the elements involved in the business process description following identification criteria. The output of this phase is an organizational model, i.e., lists for: task, job, organizational unit, resource, which includes relevant actors and their respective tasks.

One important and recurrent element that is of our interest is the concept of task. We identified [6] a set of criteria to identify the concept of: task, process and workflow. During the practical experience of using the methodology, we identify and provide a solution to the following question: How from a textual scenario a task can be identified? We looked at four dimensions surrounding the task execution (i.e., time, space, resources, and information). Any variation of any of these four dimensions, taken alone or combined, thus generates a potential identification of a new concept.

We focus on the task identification using the following identification criteria:

- Change of space (or change of location): when the scenario indicates a change of location of the operations, a change of task occurs. Therefore, any scenario fragment like "in the headquarters, the worker does . . . , and then in the local agency, the worker does. . . " indicated a change of space, therefore a change of task.
- Change of resource: when the scenario suggests that new or different resources are exploited, a change of task occurs. We distinguish three categories of resources: change of "User stereotype"; change of resource of type "material"; and change of resource of type "immaterial".
- Change of time: when the scenario indicates a different time period in which the task is performed. We differentiate four criteria: Existence of an interruption; existence of a waiting point; permanence of execution unit; periodicity of execution.
- Change of nature: when the scenario represents a change of category a change of task occurs.

In our case study, we identify the most important tasks according to the preceding, which are the following:

1. **Create Accunt.** The user registers their basic data such as name, email address, password, personal website, phone, institution, employment, birth date, hobbies.
2. **Login.** The user accesses the system to do so must enter your user name and enter your password. This task includes sub-task to recover password.
3. **Modify Account.** The user edits its profile.
4. **Upload Content.** The user enters his curriculum vitae, academic training, vocational training and employment, enter courses and seminars, conference participation (attendee or speaker), enter the language that dominates and the level you have. Also, the user can expand your basic information including a picture for your profile.
5. **Read Bulletins.** Articles available for users about news from the research community.
6. **Search.** The user can search any of his contacts, community member o any other content related search.
7. **Manage Events.** The user can add a category for events, event Location, starting time, ending Time Events, places for Events. An event can be marked as private, public, or what is even more interesting a news post is set on the landing page, making information available for the whole community. Finally users the user can view Events at any time.
8. **Manage Work Groups.** The user can create groups of contacts, add a description, a category, view advanced options. Posting the new created group to the landing page of the community is also possible.
9. **Manage Contacts.** The user edits its contact list.

At first everything is classified as tasks, and then comes the problem of *How to group tasks, processes and workflow? Where do they belong?* A task could be part of a process model or a task model. Existing knowledge on task identification criteria is again relevant to make such separation, table 1, summarizes these identification criteria.

Process Modeling

The question "What to do?" is answered with the definition of a process indicates which tasks must be performed and in what order. Thus answering the question what to do? After having identified tasks that are part of a process then they have to be related to each other by means of process operators. We propose the use of Petri Nets notation for modeling processes. As guidance for process modeling, designers must rely on: Petri Nets Structure Rules [13], Identification Criteria [6], and the WF Modeling Guideline by Example material available at (http://www.usixml.org/index.php?mod=pages&id= 40).

Table 1 .

Model Identification Criteria

	Criteria		
	Time	**Space (location)**	**User Stereotype**
Workflow	Series of time periods	Different locations	Different groups of resources
Process	Series of time periods	Same location	One resource or a group of resources
Task	Same time period	Same location	Same resource

In our case study we model the sequence of processes using the YAWL tool [13] (figure 2). In order to access to the personal page, the user must Login if he is already registered or create an Account if he is not registered. This diagram demonstrates what processes need to be executed in order to reach a determined process. For example, if the user wants to upload information, he must access to his personal web page first.

Task Modeling

The question "How to do it?" is answered with task modeling. For each task in a process a task model can be specified, not necessarily, to describe in detail how the task is performed. By exploiting task model descriptions different scenarios could be conducted. Each scenario represents a particular sequence of actions that can successfully be performed to reach a goal. Task models do not impose any particular implementation so that user tasks can be better analyzed without implementation constraints. Our task model (figure 3) represents a decomposition of tasks into sub-tasks linked with task relationships. It is an extended version of UsiXML task modeling [7] and compliant with the graphical notation of CTT [11].

Due to lack of space, we are only to show the task tree of a significant subtask, which is Manage Events (figure 3). In the top of the tree is show the subtask Manage Events which is discomposed in several subtasks, which the user can do at least one of the following tasks: Show all Events, Show my Events, Show Pending Event Invitations, Show Past Events, Search or Create New Event, then after a event is selected, the user can add a bulletin or Change the image for the event.

It is worth to say that Create New Event Task is composed by a set of different tasks which could be done concurrently, i.e. a task can be started then interrupted while another task is being done, and then continued to the task which has been interrupted, also the order of the tasks does not matter, while each one is done.

User Interface Modeling

The methodology involves a set of models that capture the various aspects required for this purpose, a UI description language to specify the corresponding UI, a methodology to structure the usage of these models, and a software support. For this purpose, a workflow is recursively decomposed into processes that are in turn decomposed into tasks. Each task gives rise to a task model whose structure, ordering, and connection with the domain model allows a semi-automated generation of corresponding UIs (figure 4).

The method proposed is applied to the automation of learning process integrating human and machine base activities, in particular those related with collaborative technology.

It is structured in a Model-Driven Architecture (MDA) paradigm, more particularly, adopting the Model-Based User Interface Development (MBUID) paradigm used for developing interfaces based on constructing a declarative description of how an interface should look and behave (model), and using the description to control the UI execution [12].

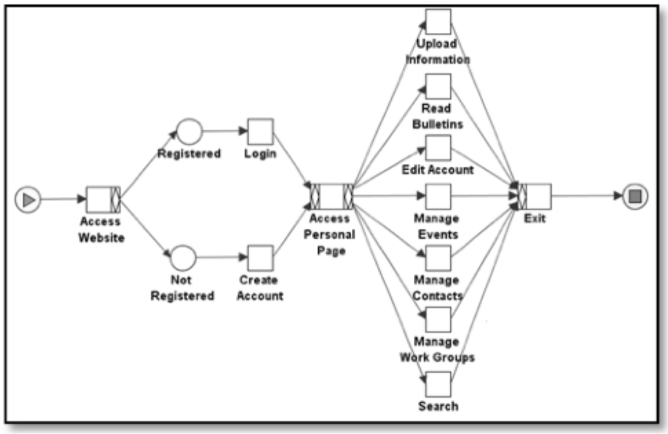

Figure 2 . Process Model of the platform.

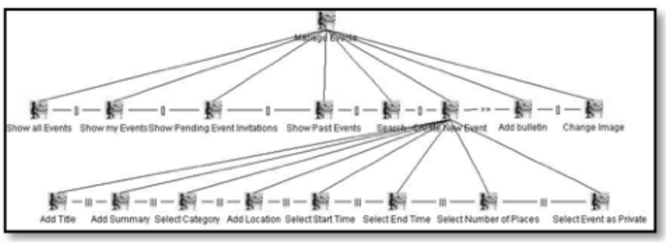

Figure 3 . Manage Events Task Tree.

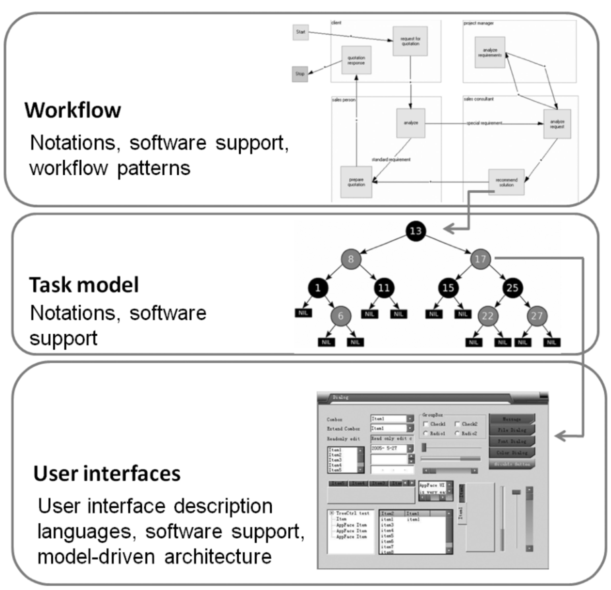

Figure 4 . An overview of the User Interface Development Method.

The Cameleon Reference Framework [2] in a simplified description, structures four development steps:

1. **Task & Concepts (T&C):**Describe the various users' tasks to be carried out and the domain-oriented concepts as they are required by these tasks to be performed.

2. **Abstract UI (AUI):**Defines abstract containers and individual components, two forms of Abstract Interaction Objects by grouping subtasks according to various criteria, a navigation scheme between the containers and selects abstract individual component for each concept so that they are independent of any modality. An AUI is considered as an abstraction of a Concrete User Interface with respect to interaction modality. At this level, the UI mainly consists of input/output definitions, along with actions that need to be performed on this information.

3. **Concrete UI (CUI):**Concretizes an abstract UI for a given context of use into Concrete Interaction Objects (CIOs) so as to define widgets layout and interface navigation. It abstracts a final UI into a UI definition that is independent of any computing platform. Although a CUI makes explicit the Look & Feel of a final UI, it is still a mock-up that runs only within a particular environment. A CUI can also be considered as a reification of an AUI at the upper level and an abstraction of the final UI with respect to the platform.

4. **Final UI (FUI):**Is the operational UI, i.e. any UI running on a particular computing platform either by interpretation or by execution.

The user interface design processes starts with a task model that is processed through an incremental approach to the final UI (figure 4 shows the four levels that are involved in the design of a UI using the Cameleon Framework). In figure 5 we show an example of the User Interface Development Method by our case study, and the task of create event which is already described before. At the top of the task tree is the main subtask, which is Manage Events, this tasks corresponds to main menu, then when it is clicked it is shown the subtasks of the task Manage Events like Show all events or Create a New Event, these are set at the second menu, which is for the subtasks of a particular task, in this case we chose the create menu task, which displays a form where the user interact with the system by performing all subtasks of Create New Event task. These subtasks can be done in a concurrent way, which means that the user can begin by adding the title, stop and then adding summary, stop and continue with the tasks, until every task is finished.

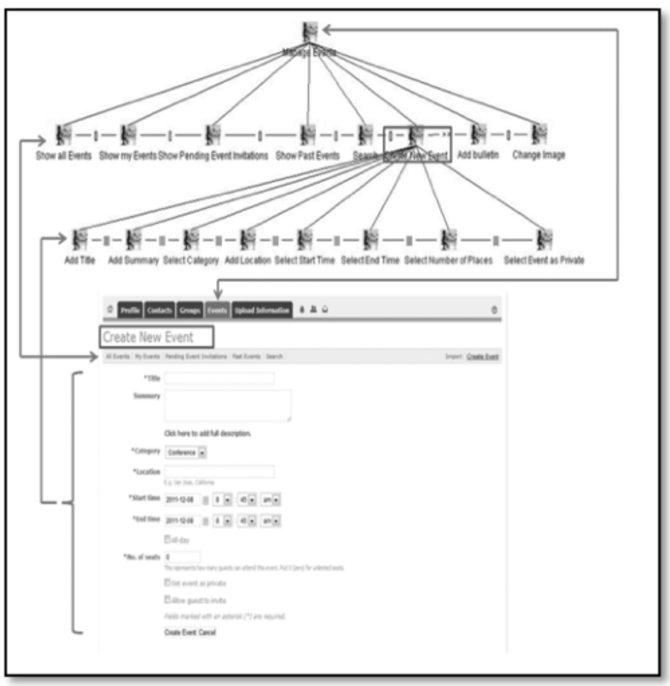

Figure 5 . Task Tree Maping into a User Interface.

EVALUATION

Designers and developers were asked to express how they would perceive the benefits of the proposed solution. They were told to look at the tool, listened to a description on how to use the system, where the various steps of the collaborative space were explained and exemplified. If additional information was requested, the written material was made available to them. The anonymous designers belong to the faculty of Computer Science of the University of Puebla. After viewing the material, designers were asked to accomplish a series of tasks, see figure 2. After that they had to fill in as honestly as possible the IBM Computer Satisfaction Usability Questionnaire (CSUQ). This questionnaire was selected because of its high reliability, its simplicity, and its high correlation with the results (empirically proved with r=0.94) [11].

This questionnaire is decomposed into 19 questions that are structured in four groups: system use (SYSUSE-Q1 to 8), information quality (INFOQUAL-Q9 to 15), interface quality (INTERQUAL-Q16 to 18), and overall estimation (OVERALL-Q19). Each question is answered on a 7-point Likert scale.

Each question is evaluated in a range of seven points, which seven is the best and one the worst. Then we obtain the average from each group and the standard deviation, whose allow us to know the range of qualification we obtain in each category. Although the amount of tested participants (8) is not considered as statistically significant, table 2 certainly gives some indication where the benefits are potentially perceived by designers: the worst score is related to documentation (Avg = 4,88), which is understandable since designers did not have enough time to review the material.

The best score was obtained with the system interface (Avg = 6,06), which may suggest that designers tend to perceive more the benefits of the methods through the software tools that support the method than the method itself. The system interface was perceived good, and so was the pleasure to use the system. This does not necessarily mean that the method is actually structured and pleasant to use, but that it is perceived subjectively by designers like that, which seems important to identify. In table 2 we present the values we obtained from our system. The High and Low values are obtained by adding or subtracting the average the standard deviation respectively. Although the amount of tested participants (8) is not considered as statistically significant, table 2 certainly gives some indication where the benefits are potentially perceived by designers: the worst score is related to documentation (Avg = 4,88), which is understandable since designers did not have enough time to review the material. The best score was obtained with the system interface (Avg = 6,06), which may suggest that designers tend to perceive more the benefits of the methods through the software tools that support the method than the method itself. The system interface was perceived good, and so was the pleasure to use the system.

This does not necessarily mean that the method is actually structured and pleasant to use, but that it is perceived subjectively by designers like that, which seems important to identify.

Question ID	Question statement
1	Overall, I am satisfied with how easy it is to use this system
2	It was simple to use this system
3	I can effectively complete my work using this system
4	I am able to complete my work quickly using this system
5	I am able to efficiently complete my work using this system
6	I feel comfortable using this system
7	It was easy to learn to use this system
8	I believe I became productive quickly using this system
9	The system gives error messages that clearly tell me how to fix problems
10	Whenever I made a mistake using the system I recover easily and quickly
11	Information (such as online help, on-screen messages and other documentation)
12	It is easy to find the information I needed
13	The information provided for the system it is easy to understand
14	The information is effective in helping me complete the tasks and scenarios
15	The organization of the information on the system screen is clear
16	The interface of this system is pleasant
17	I like using using the interface of this system
18	This system has all the functions and capabilities I expect it to have
19	Overall, I am satisfied with this system

Table 2 .

CSUQ Results.

	System's use	Documentation Quality	Interface Quality	General Value
High	6, 1817	5, 9029	6, 5599	6, 4324
Average	5, 9375	4, 8857	6, 0666	5, 8
Low	5, 6932	3, 8684	5, 5733	5, 1675

The best score was obtained with the system interface (Avg = 6,06), which may suggest that designers tend to perceive more the benefits of the methods through the software tools that support the method than the method itself.

The system interface was perceived good, and so was the pleasure to use the system. This does not necessarily mean that the method is actually structured and pleasant to use, but that it is perceived subjectively by designers

like that, which seems important to identify. In table 2 we present the values we obtained from our system. The High and Low values are obtained by adding or subtracting the average the standard deviation respectively.

Although the amount of tested participants (8) is not considered as statistically significant, table 2 certainly gives some indication where the benefits are potentially perceived by designers: the worst score is related to documentation (Avg = 4,88), which is understandable since designers did not have enough time to review the material.

The best score was obtained with the system interface (Avg = 6,06), which may suggest that designers tend to perceive more the benefits of the methods through the software tools that support the method than the method itself. The system interface was perceived good, and so was the pleasure to use the system.

This does not necessarily mean that the method is actually structured and pleasant to use, but that it is perceived subjectively by designers like that, which seems important to identify.

Although several individual questions were ranked at a medium stage, the overall satisfaction was perceived higher (Avg = 5,8), which largely contrasts with some specific issues such as: method difficult to use, information not obviously available, but method that produces acceptable results and in an organized way.

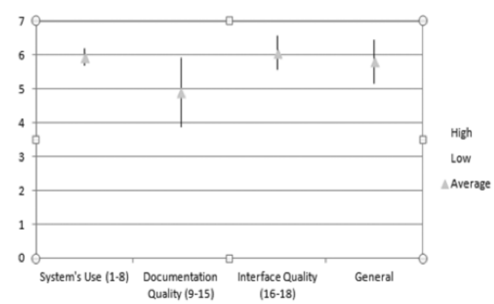

Figure 6 . Task Tree Maping into a User Interface.

This seems to be confirmed with the four respective global scores produced by the IBM CSUQ (figure 6): the system usage is estimated easy to use but not the information quality, but the interface quality was estimated better (with smaller variations) and the overall quality.

From this results we conclude that, despite we have a good score in general, we must improve the documentation quality, which qualifications were diverse as we can see in figure 7, which corresponds to the graphic of table 2.

CONCLUSIONS AND FUTURE WORK

In this paper we introduced a formal method to develop collaborative spaces. The design and engineering knowledge to successfully create those systems is documented using FlowiXML Methodology. The selection of this methodology was based on our experience in applying it to successfully address different problems. It was a challenge to use it to design of an Information System. FlowiXML designing process is focused on the user needs thus is user-centered. The development steps and notations foster discussion within the stakeholders of the problem before developing the system. The benefits of the resulting system were evaluated using the IBM CSUQ questionnaire. The results are promising as the acceptation of the produced system was perceived high. Still there is work ahead to produce more robust system including the design of collaboration and how to motivate it through the user interface.

ACKNOWLEDGMENT

Josefina Guerrero García and Juan Manuel González Calleros acknowledge CONACYT program "Repatriación" to support this research.

REFERENCES

[1] Bracchi, G., Pernici, B. (1984). The Design Requirements of Office Systems.*ACM Trans. on Office Information Systems,* 2(2).

[2] Calvar Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J. (2003). A Unifying Reference Framework for Multi-Target User Interfaces. *Interacting with Computers,* 15(3):289-308.

[3] Currin, L. (2003). Feelin'groovy: Experts now believe that e-learning must elicit positive emotions to succeed. *eLearn Magazine.*

[4] Georgiev, G., Dimitrova, T., Karamanska, D. (1988). Ergonomic Factors of Computer-Assisted Learning. *VMEI Sofíam.*

[5] Guerrero, J., Vanderdonckt, J., Gonzalez Calleros, J.M. (2008). FlowiXML: a Step towards Designing Workflow Management Systems. *Journal of Web Engineering,* 4(2):163-182.

[6] Guerrero, J., Vanderdonckt, J., Lemaigre, Ch.(2008). Identification Criteria in Task Modeling. *Proc. of 1st IFIP TC 13 Human-Computer Interaction Symposium HCIS'2008,* 272:7-20.

[7] Guerrero, J., Vanderdonckt, J., Lemaigre, Ch.,González Calleros, J.M. (2008). How to Describe Workflow Information Systems to Support Business Process. *10th IEEE Conference on E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing,* 404-411.

[8] Guerrero García, J., Lemaigre, Ch., Vanderdonckt, J., and Gonzalez Calleros, J.M. (2008). Model-Driven Approach to Design User Interfaces for Workflow Information Systems.*International Journal of Universal Computer Science,* 14(19):3236-3249.

[9] Hilera, J. R., Palomar, D. Modelado de procesos de enseñanza aprendizaje reutilizables con XML, UML e IMS-LD. *Revista de Educación a Distancia, número monográfico II.* http://www.um.es/ead/red/M2/.

[10] Kort B., Reilly R., Picard R.W. (2001). An Affective Model of Interplay Between Emotions and Learning: Reengineering Educational Pedagogy-Building a Learning Companion. *Proceedings of International Conference on Advanced Learning Technologies (ICALT 2001).*

[11] Lewis, J.R. (1995). IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and In-structions for Use. *International Journal of Human-Computer Interaction,* 7(1):57-78.

[12] Puerta, A. R. (1997). A model-based interface development environment. *IEEE Software,* 14(4): 40-47.

[13] van der Aalst, W.M.P. (1998). The Application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers,* 8(1):21-66.