THE INCREMENTAL DESIGN OF SCRIPTS BASED ON MULTI-AGENT SYSTEM

Sara Boutamina
Mentouri University, B.P. 325 Route Ain El Bey, Constantine 25017, Algeria

Hassina Seridi
LabGed laboratory, Badji Mokhtar University, BP 12 Annaba 23000, Algeria

Abdelkader Gouaïch
LIRMM Laboratory, MontpellierII University Montpellier, France

Keywords: Collaboration Scripts, Tracking Learners, Traces, Design Method, Multi-Agent Systems.

Abstract: Collaborative learning is not always effective; its effects depend on the richness and intensity of interaction between students during the collaboration (Dillenbourg, 2002). This collaboration is structured using collaborative scripts. Hence, the design of these Scripts is not trivial; it requires information on learners and on their interaction. We believe that when learners are the target of any design, this one needs to be evaluated on the basis of the learners themselves. However, most of the design approaches do not use experimental feedback on the learners' collaboration to improve the initial design. We propose in this article a method for the design of scripts basing on the experimental feedback. We suggest the use of multi-agent systems to provide help and information to the scripts designers.

1 INTRODUCTION

These recent years, researchers stress more the importance of learning design. Among these researchers, Robe Koper and Tattersall (. Koper & Tattersall, 2005) who state that "the key principle in learning design is that it represents the learning activities and the support activities that are performed by different persons (learners, teachers) in the context of a unit of learning".

Koper thinks that the key of the success of the learning environments is the activities and not the pedagogical objects. Consequently, he proposes, to specify the learning situations, the Educational Modelling Language (EML) which focuses on the pedagogical activities. This language was adopted by the IMS Global Learning Consortium to propose the standard IMS Learning Design (IMS LD).

The result of the design of the learning situation is called a script which is considered as a sequence of phases.

In our work, we affirm that this concept (script) is linked to the concept of trace. This later can contribute to the changing of the script, either in a dynamic way in order to regulate the learning, or at the end in order to evaluate and reuse this script.

In this paper, we propose an approach for the construction of scripts taking into account the experimental feedback on the learners' collaboration. The idea is to track the learners when performing the different activities prescribed by a script which is designed at first (preliminary design) and provide feedback on the execution of this script in order to review the preliminary design).

2 OBJECTIVE AND MOTIVATION

We A collaborative script (or scenario) is a set of instructions prescribing how students should form groups, how they should interact and collaborate and

Boutamina S., Seridi H. and Gouaïch A. (2009).
THE INCREMENTAL DESIGN OF SCRIPTS BASED ON MULTI-AGENT SYSTEM.
In *Proceedings of the First International Conference on Computer Supported Education*, pages 320-325
DOI: 10.5220/0001989303200325

how they should solve a problem (Dillenbourg, 2002). It structures the collaborative process in order to promote specific types of interactions (Dillenbourg, 2006 (a)). A script includes multiple activities, occurring at different various social levels (Dillenbourg, 2006(b)): individual activities (e.g. reading, writing…), group activities (e.g. solving a problem with a peer…), and class wide activities (lecturing, discussion…).

A variety of design methods of scripts have been proposed but none of them take into account the experimental feedback and use it in an incremental way in the process of scripts design. In fact, these methods rarely use the feedback to improve incrementally the initial design and most of them focus more on the results of collaboration rather than the process of design itself.

Also, designers have to take in consideration the learners and their behaviours because they are at the end the main actors of the designed script.

Our framework of scripts design is based on the following six ideas:

1. The process of design is incremental based on a loop of four phases which are: Scripting, Specification, Execution and Evaluation (Fig 1.).

Scripting is the phase of writing, for a group of learners, of the different rules of collaboration and describing the different activities, the different roles, etc. In this phase a natural language can be used.

In the phase of specification, the script is specified using a specific formalism. Then, this script will be executed and finally it will be evaluated on the basis of the learners' traces during the scripts execution.

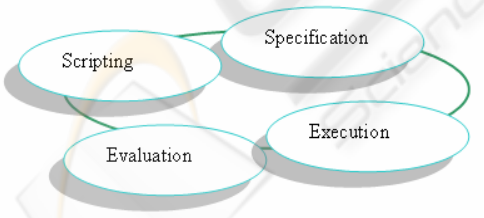

Figure 1: The different phases.

2. The Scripting must be considered as a whole and not only through its outcome i.e. we have to take into account:

- The starting point (the different data).
- The final point (the outcome).
- The transformation from the first point to the second one.

3. A design method is to guide without any constraints the designer who must take into account the human factors involved in the execution of the script. The execution model of the script must be considered as a task or activity model and not a data model.
4. Human factors play a central role in the process of design. Designers require information on learners and on their collaboration in order to favour the desirable interactions. For this reason we suggest the tracking of the learners.
5. Formalism for the specification of scripts is used. This formalism enables the designer to express his choices and not only to describe the result.
6. An integrated environment is desirable for the scripts design in order to facilitate continuous communication between the various "activity spaces" of the design process. Hence, in an incremental approach of scripts design the designer can move from the evaluation spaces to the specification spaces and to the implementation spaces. We propose that the use of such environment provides a way to overcome the problem and gives designers tools to go beyond the assumptions of standard design.

3 THE INCREMENTAL DESIGN OF SCRIPTS

The developer of pedagogical scenario can not judge a design choice only if he evaluates its consequences in a real situation based on the feedback of the learners interactions. Also, we recommend an iterative process in which the results of the developed scenario evaluation are analyzed and interpreted in order to be used for the adaptation or for the improvement of the scenario.

The execution of the script should be considered as a task or activity model and not as a model of the different resources offered to the learners.

- The idea is to allow the designers to express their choices and not only describe the script. Indeed, the script is the expected result but some choices may be important as they are represented. For instance they can be used in the reuse and adaptation of scripts.

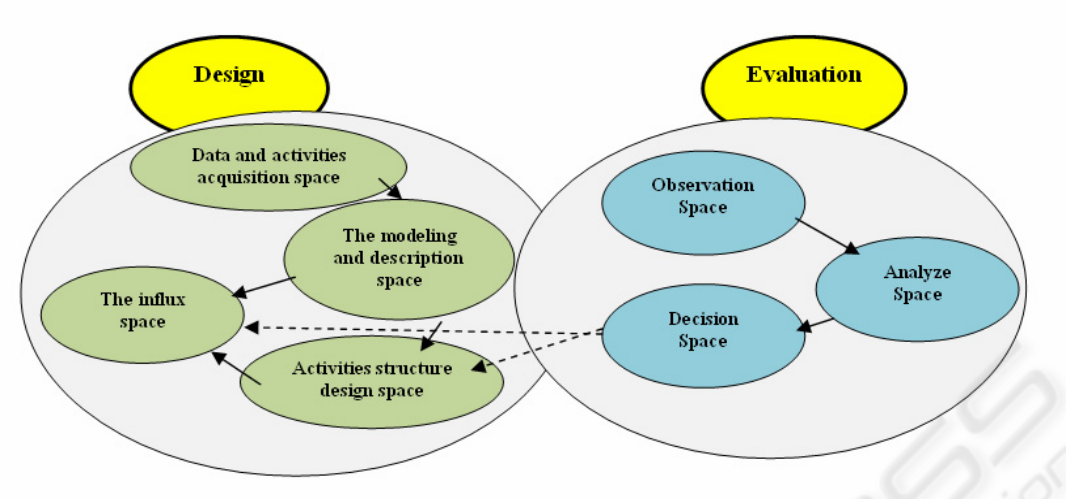

Figure 2: The different activity spaces.

- To facilitate the proposed design work, an integrated environment is essential to allow the designer to move between different areas of activity. Indeed, the designer is evolving from assessment space, to specification and implementation spaces.
- The constraints of such approach are the automatic generation of script that can be possible only by a formalization of the outcome of the design and the use of a set of artificial agents that will act in the different activity spaces listed below.

What Activity Spaces and how to Skip from an Area to Another?

The steps concepts must be distinguished from that of space activity in a design. Indeed, a stage characterizeed a specific product and design methods are described in terms of stages. The activity space characterizes a state of the developer's activity. We have identified and characterized the different activity spaces of the design process. These spaces and the links between them are presented in the following figure (Fig.2.).

We can identify seven activity spaces useful for the designer over two phases: the design and evaluation. Each activity space represents an identifiable viewpoint of the designer on its design task.

For the design phase, we have identified four areas of activity spaces:

- *Data and activities acquisition space* in which information is collected: pedagogical resources, profiles, learning activities...
- *The modeling and description* space: allows the designer to have key abstractions and a clear vision and accurate information it will use. This is similar to application development approaches.
- *Activities structure design space:* the designer main concern is developing a model of activities or pedagogical resources stemming from the task for which the script is designed. This model is different from a data model.
- *The influx space:* provides the designer with the means to specify how to change the specification of the script structure to the instantiated structure of the target script.

For the evaluation phase, we suggest the following three areas:

- *Observation Space:* The main concern is to observe the progression of learners in accord with their requirements and profiles.
- *Analyze Space:* Actions and interactions of the learners in the groups are analyzed from the observation delivered in the precedent space in order to have synthesized information about the learners' progression in the group and about the designed script.
- *Decision Space:* The main concern is to have some decision about the script to be presented to the script's designer in order to refine this later.

4 THE CONCEPTUAL MODEL OF MULTI-AGENTS BASED SYSTEM FOR THE SCRIPTS DESIGN

Jennings and his colleagues (Jennings et al., 1998) argue that the use of agents is attractive because they are able to characterize naturally and easily a variety of applications, and also to represent the different entities of a system or a domain.

The agent paradigm is the most powerful paradigm to provide abstractions for complex organizations analysis and modelling. Humans and software systems can be considered as entities which interact and collaborate to perform their tasks in order to achieve their goals.

In order to run a successful multi-agents simulation in this approach, a script (scenario) must be provided to agents. A collaborative script differs from a program in that no explicit specification is given in advance. It is necessary to propose a conceptual model that models agents at an appropriate level of abstraction by executing the script and indexing actions and interactions of learners in the learning environment.

The script author and an agent developer agree upon activities as the interface between them. The script author describes scripts using a language, while the agent executor implements the activities to be performed by learners and extracts interactions from the script. The script author describes scripts Using the learners' activities and interactions in the group.

The script executor conducts experiments in a real environment, and then the experiment outcomes are used by the script writer in order to refine the original script.

The evaluation agent observes learners interactions by collecting different traces of learners during their collaboration.

5 AGENT FOR PROBLEM AREA

Learners are different and it is difficult to have an adequate script for the entire group from the beginning. In order to help the designer to modify his script on the basis of learners, we suggest the use of a set of agents having the following roles: decision, interpretation, execution, observation and tracking learners.

The work of the agents starts when the different learners interact with the system.

A Graphical User Interface (GUI) is used to facilitate the learners' interactions with the system. Each learner has to introduce his profile using this interface. These profiles are stocked in the '*Profiles Base'* by '*The Decision Agent'*. This agent has a direct relation with '*The Interpretation Agent'* which has to specify the script in a comprehensible format for the other agents. This script is executed by '*The Execution Agents'*.

'The Tracking Agents: keep track of the learners and stock the different traces in '*the Traces Base'*. '*The Observer Agent*' controls the works of the other agents.

The Decision Agent: Basing on the learners profiles, the decision agent selects a script (which is adequate to the profiles of the learners and not to the behaviours/collaboration of learners) to be executed in order to structure the learners' collaboration.

Moreover, this agent is able to access directly to 'the Profiles Base' and 'the Traces Base' in order to provide the designer with the necessary information.

The Interpretation Agent: The script is specified using a format which is different from that used by agents; consequently, this script will be interpreted by *'the Interpretation Agents'* to make it comprehensive.

The Execution Agent: '*The Execution Agents'* execute the interpreted script taking into account the different learners' profiles.

The Observer Agent: The execution of the script will be controlled by *'the Observer Agents'*. These agents monitor the work of the other agents to provide general information on the execution of the script.

The Tracking Agent: They collect the different traces of the learners during their collaboration.

6 A CASE STUDY ON THE ARGUEGRAGH SCRIPT

The ArgueGragh (Dillenbourg, 2002) script is a macro script aimed to trigger argumentation between peers. It consists of the following five phases:

1. Each learner responds to an on-line multiple choice questionnaire and for each answer he is expected to argue his/her choice.
2. When all the learners answer the questionnaire and argue, the system produces a corresponding graph where the learners are positioned according to their answers. Then, the teacher or the system forms pairs of learners who provided different answers in Phase 1.

3. Each pair has to respond to the same questionnaire in Phase 1 and provide arguments. They can see their answers and justifications provided by each peer in Phase 1.

4. For each question the system calculates the answers given individually and in collaboration. The results are used in a debriefing session where learners comment their arguments.

5. Each student writes a summary of all the arguments collected for a specific question. The summary should be structured according to the framework used in the debriefing session.

Application of the Proposed Model to the ArgueGraph Script

When the learners are present, the decision agent informs the interpretation agents in order to rewrite the script in a comprehensible format for the other agents. Then, this script is executed by a set of execution agents. These agents provide the learners with the questionnaire and each learner responds to it and argues his/her choice.

According to the learners answers the decision agent produces the corresponding graph. Then, the teacher or this agent forms pairs of learners who have conflictual answers.

Each pair has to respond to the same questionnaire in Phase 1 and provide arguments. The decision agent allows the learners to see their answers and justifications in Phase 1.

This agent calculates for each question the answers given individually and in collaboration. The results are used in a debriefing session where the learners comment their arguments.

The different interactions of the learners with the system are collected by the tracking agents and during all these phases, the observer agents monitor the other agents in order to provide a general idea on the execution of the script.

In this way the designer can modify his script and adapt it on the basis of the learners assisted by a set of artificial agents which gave him the necessary information about the learners' interactions and actions.

6 THE AGENTS IMPLEMENTATION

To allow learners to access the learning system, a distributed learning environment is proposed for learners located anywhere and connected to learn at any times. It's a multi-agent based distributed learning environment which provides a multitude of learning object for learners of the group which are referenced by the script author.

The learning system consists of the client side and the server side. On the client side it has a JSP (Java Server Page) user interface. On the server side, the servlets and a multi-agent platform implemented using JADE (jade: http://jade.tilab.com).

JADE (Java Agent Development Framework) is a software framework for the development of multi-agent systems and conforms to the FIPA specifications (fipa : http://www.fipa.org/).

When learners log on the system through Web based applications, a learner agent upload the profile and requirements and the learner is affected to the assigned group. The script is uploaded and the execution of the script will be performed.

7 CONCLUSIONS

Collaboration has certain advantages for learning. To profit from these advantages, the learners' collaboration should be structured and organized. Hence, scripts are used to structure the desired interactions among learners.

The design of these scripts in not easy, for this reason we suggest the use of an incremental script to help the designer to take into account the behaviours of learners and their interactions.

In this paper we presented a multi-agent based system for the incremental design of collaborative scripts. The main agents of this system are, namely, '*The Decision Agent', 'The Interpreter Agents', 'The Execution Agents', 'The Tracking Agents'* and *''The Observer Agents. These agents have* the following roles: decision, interpretation, execution, observation and tracking learners.

REFERENCES

Dillenbourg, P, 2002. Over-scripting CSCL: the risks of blending collaborative learning with instructional design. In P. A. Kirschner(Ed) *Three worlds of CSCL. Can we support CSC,* Heerlen, Open UniversiteitNed. pp: 61-91. Netherland. 2002.

Dillenbourg, P, 2006(a). The solo/duo gap. *Computers in Human Behavior*. Elsevier Ltd. 22, pp. 155- 159.

Dillenbourg, P, 2006(b). Orchestrating integrated learning scenarios.*Proceedings of the 23rd annual ascilite conference: who's learning? whose technology?* 2006.

Foundation of Intelligent Physical Agents (FIPA), http://www.fipa.org/

Java Agent Development Environment, http://jade.tilab.com/

Jennings, N. R, Sycara K. P, and Wooldridge, M, 1998. A roadmap of agent research and development. *Journal of Autonomous Agents and Multi-Agent Systems,* 1(1): pp. 7–36.

Koper, R, and Tattersall, C, 2005. Learning design: a handbook on modelling and delivering networked education and training .Berlin; New York: Springer.