Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Adaptive Planning Content Sequence with Workflow

Robson da Silva Lopes
Computing Department
Federal University of Uberlândia
Uberlândia, Brasil
robsonsilvalopes@hotmail.com

Márcia Aparecida Fernandes
Computing Department
Federal University of Uberlândia
Uberlândia, Brasil
marcia@ufu.br

***Abstract*—Actually, the most important challenge of distance learning environments is taking into account the specific and particular characteristics of each student. A dynamical accompaniment of students, during the course, allows one to consider these aspects and consequently adapts the content presentation to learning style, behavior, rhythm and knowledge of each student. This problem has been divided in two parts. The first one is the content sequencing, that has been successfully accomplished by AI techniques, specially AI Planning. The second one is the adaptation of content sequence presentation to the student profile that needs a student model. So, in order to cope with this problem, this work presents an approach based on AI Planning techniques and Workflow for generating content sequences. The Workflow technology has great similarities with Planning and allows the management of student activities. The student model considers the learning style, profile and preferences of each student and in this way it is possible to adapt the content sequence presentation to the student.**

***Index Terms*—Distance Learning Environments, learning style, workflow.**

I. INTRODUCTION

Interaction and cooperation between tutor and student are important aspects in the learning/teaching process, in that it enforces the necessity to consider the student as an individual with his own characteristics. Therefore, mechanisms that provide a learning process in this perspective should be considered.

Technologies for the development of Distance Learning Environments (DLE) have integrated resources and techniques that rise from different areas of knowledge such as Education, Computation, mainly Artificial Intelligence (AI), Hypermedia, Cognitive Psychology and Pedagogy, in order to allow such environments to adapt themselves to the student characteristics and education objectives.

An important aspect in DLE for presenting intelligence and adaptivity is the automatic sequencing of content. Some attempts to resolve it have used AI Planning, which in this case is called the Instructional Planning.

However, besides supporting adaptation to the student learning style, there is a great possibility of monitoring the contents and the student during the course. The coordination, monitoring and implementation of the course become preponderant tasks to be considered during the planning and execution of a course [1], because it involves a dynamic environment and with a high degree of interaction between the participants of the learning process.

Therefore, due to such characteristics, the Workflow modeling techniques have been submitted as a possibility for accomplishing these tasks in a distance learning environment by allowing the definition of the teaching process as a whole, with the clear specification of all of the activities to be executed, the relationships among these and who should execute them.

Given the similarities between AI Planning technique and Workflow, it becomes possible to associate the two techniques for generating customized content, taking into account the student profile and the learning style, as well as for monitoring the environment of education.

So, this article presents an approach for content sequencing based on AI Planning and Workflow technology, that allows the DLE to adapt the instructional content to the preferences and learning style of each student.

This paper is organized as follows. Section 2 presents a workflow planner based on AI Planning techniques and genetic algorithms and describes the modeling of a course by using this planner. Section 3 defines the student model and shows how the adaptivity is integrated to the content sequencing taking into account this model. Section 4 presents a case study related to a Java course. Section 5 presents the conclusions and further works.

II. AI PLANNING AND WORKFLOW FOR CONTENT SEQUENCING

The main similarities stressed in works involving these two areas are related to activity sequencing aspects and the presence of agents for the execution of same. Both approaches produce a route for the generic instance solutions of a particular problem. In workflow systems, this route is a list of activities and in Planning, it is a sequence of actions. In Workflow, the aim is the solution of a case, which may be solved by instancing a given process.

Workflow technology is coming as an alternative in modeling and managing DLE activities, because it allows the generating of different content presentations and the monitoring of the course, making monitoring of the student by the tutor easier [1].

In the context of the DLE, if the courses are defined as Workflow, the Workflow participants are tutors and students, the activities (actions) correspond to the instructional activities of the student, such as, exercises, learning, research, and so on.

In some works Workflow technologies have been used in DLE such as Flex-el [2] and Virtual Campus [3] which

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

use Workflow in the management of learning activities and assessment process of a course between tutor and student. Pereira[4] presents a work in which Workflow technology aids content sequencing and presentation, and the management of the whole DLE, such as coordinating matters or educational issues [5], [6].

In Alves et al [7] a general planner (Workflow Planner) based on genetic algorithms and AI Planning techniques that generate a workflow for a set of actions previously defined and published in a repository is presented. So, this planner is used in the first stage of the instructional planning in order to obtain a workflow of the content sequences. The next subsections present a general description of this planner [1] and the application to a domain of Java Course.

A. *Workflow Planner*

The correspondences between Planning and Workflow may be seen in Table I in which one observes that a process instance when executed (case), is considered as a plan in Planning. The execution of an activity in Workflow is seen as an action in Planning. A workflow resource (persons, applications, machines) is seen as an executive agent in Planning. This mapping allowed the application of Planning techniques to some stages of the workflow system, guiding the effort of subsequent researches.

TABLE I
PLANNING AND WORKFLOW MAPPING

Workflow	Planning	Instructional Planning
Case	Plan	Content Sequence
Activity	Action	Instructional Activity
Resource	Agent	Tutor/Student

The workflow planner architecture was depicted in Figure 1. The first stage Activity/action mapping (2) consists of receiving the available information in relation to the activities (1) in the format XPDL(XML Process Definition Language) and generating the equivalent in PDDL (Planning Domain Definition Language). The description of the activities in the repository (1) of the system may be performed manually or through any addition tool that contemplates the format XPDL and fulfills WfMC[2] specifications.

Based on the production rules of the published activities, the GeneticModeling module (3) generates a preliminary model containing a unified representation of possible combinations of process models (4). In generating these combinations, all activities capable of being unified, in other words, all activities which pre and post-conditions are fulfilled at some point of the process are considered in the representation. The GeneticModeling allows one to consider not only conditional and sequential rotation, but also the use of parallel rotation. All these rotations are treated at once due to the use of the paradigm genetic algorithm and a planner of general objective, once it is made possible by considering problems of several domains of the real world.

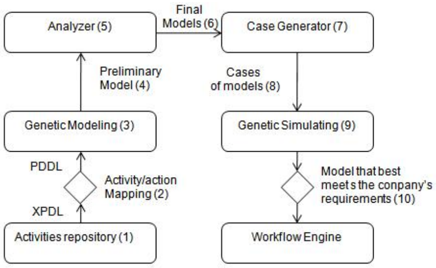

Fig. 1. Workflow Planner Architecture

Activities or blocks of activities that play the same role (conflicting points) indicate the existence of more than one process model for the same problem. Thus, the Analyzer (5) becomes responsible for extracting the final models, in other words, the possible valid process models. The Analyzer extracts the final models already considering conditional (or), parallel (and) and sequential activities (seq). Each valid process model (6) is manipulated by the Case Generator (7), whose functionality is to generate (8) the possible existing cases, in other words, the possible routes in the process model. In the situation where the case generator finds an OR-type activity rotation, relative instances for each one of the possible alternative routes will be generated. For the AND-type rotation, a serialization of these activities is performed and instances that alternate the execution order of these activities are generated. For the other cases where the activities of the process model are sequential, the work mapping is direct. The cases are sent to the Genetic Simulating module (9) that, along with information obtained from recourses associated to each activity and its respective execution time, performs an analysis of the models that best meet the company's current requirements. The return of the analyzed model is also converted into format XDPL aimed at its execution (10) in a Workflow motor that contemplates such representation. A preliminary model of all valid processes, all activities which pre-condition of the origin activity are fulfilled with the post-condition of the destination activity considered. The development of the population is performed through the enlargement of genetic operators.

This work uses a genetic planner that is characterized by performing a global search in the solutions space, unlike classic planners that perform this search locally. This characteristic allows genetic planners to manipulate large-dimension problems. Another important characteristic is a flexible representation that allows one to consider aspects such as conditional and parallel flow between activities that have not been explored much in previous works and that are important when one considers modeling processes. The genetic algorithm evolution is determined by using planning techniques. An analysis of the transitions of model activities is performed in order to certify

---

[1] A detailed description related to genetic algorithm for this work can be seen in Alves [7]

[2] WfMC - Workflow Management Coalition is a global organization of adopters, developers, consultants, analysts, as well as university and research groups engaged in workflow and Business Process Modeling

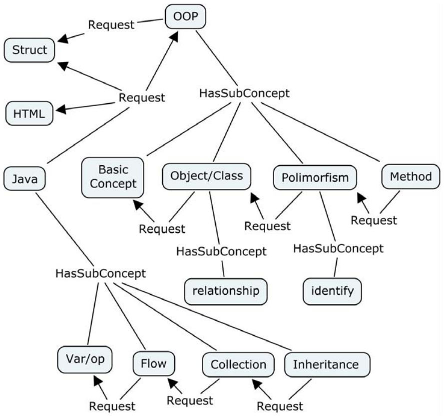

Fig. 2. Conceptual map of a Java Course

if the pre-conditions of the origin activities have been fulfilled in the post-conditions of the destination activities.

B. Course Modeling

Course construction considers the use of conceptual maps, since this formalism has been used as a didactic resource and it is well-known in the educational context, different from Workflow and IA Planning techniques. The conceptual maps are based on the cognitive theories and are a structure for representing the concepts of a content and relationships among these through hierarchical diagrams and seek to reflect the organization of an individual cognitive structure in a given theme.

In this work, the present concepts in a conceptual map are classified in: Extensive Concepts and Sub-Concepts. Extensive concepts are defined as every concept that contains a sub-concept. A sub-concept is a concept that is part of an extensive concept and which does not contain sub-concepts. In other words, the extensive concepts may be parts of another extensive concept, but they are not sub-concepts since they do not contain sub-concepts.

In general, the design of a conceptual map should not be standardized or restricted. However, this work proposes two types of relationships that can exist between concepts and sub-concepts of a same hierarchical level: 1 - *HasSubConcepts* show that a concept destiny is a sub-concept of the concept source and 2 - *Requests* show that the concept source needs the concept destiny.

Figure 2 shows the conceptual map of a Java course graphically. In this conceptual map, *JAVA* is an extensive concept, it has the sub-concepts *Var/op*, *Flow*, *Collection* and *Inheritance*. For instance, due to the relationship *Requests* among the concepts *JAVA* and *HTML*, *JAVA* can only be learned after the student has learned the *HTML* concept. This also can be observed in relation to the *JAVA* and *Struct* concepts.

Each concept or sub-concept in the conceptual map is linked to various Learning Objects (LO) in a repository. The connection between the concept and the LO is achieved by means the category *Classification* that is defined by the IEEE LOM standard [3][8]. Thus, the LO are saved a specific LO repository. However, the system can search LO in an other repository if it follows the IEEE LOM standard [8].

Initially, the concepts of a course are represented in a conceptual map. The tool ”CmapsTools” has been used in order to build this representation, because it allows one to export a conceptual map to a XML file. Among the components of this file, the components <*concept-list* >, <*linking-phrase-list* > and <*connection-list* > are the most important for the workflow planner.

- <*concept-list*> : Contains a list of all the concepts present of the conceptual map. Each concept has a identifier;
- <*linking-phrase-list*> : contains a list of all relationships in the conceptual map. Each relationship has an identifier;
- <*connection-list*> :contains the connections between relationships and concepts of the conceptual map;

The file contains others components, those are regarding the layout and the graphic representation of the conceptual map. The XML file is imported to DLE, where the instructional actions are generated.

Instructional actions are tasks that will be performed by Workflow Management System (WFMS[4]) so that the student can learn a given concept. Each one of these actions is an Instructional Abstract Action (IAA) or Instructional Primitive Action (IPA). IAA are the actions that belong to the conceptual level, in other words, they determine that concepts will be learned with such action. There are three types of IAA as described as follows.

- Learn(): Responsible for the concept introduction through instructional activities such as introduce or present;
- learnconcept(): Responsible for identifying the end of the concept learning;
- learnSubConcept(): Responsible for the sub-concept introduction;

An IPA defines the way that the concept will be presented to the student. For instance, according the student learning style, the concept should be presented by means of an example. In this case, the IPA *toExemplify()* searches for a LO that contains examples of the concept. In this work eight types of IPA are considered.

- toDefine() - Defines the concept
- toExemplify() - Shows examples of concept
- toExplain() - Explains the given concept
- toComplement() - Provides complements for the concept comprehension
- toExercise() - Presents exercises related to the concept
- toSummarize() - Presents a summary of the concept
- toEvaluate() - Evaluates the concept learning
- toIntroduce() - Provides a concept introduction

One can observe that an IPA needs the concept (or sub-concept) and the student model in order to determine the more

---

[3] LOM - Learning Object Metadata

[4] A WFMS defines, creates and manages the execution of workflow through one or more workflow engines, which interpret the process definition, interact with workflow participants and, when requested, invoke Information technology tools and applications.

Algoritmo 1 : Instructional Action Generator

Input: R: Concept Set

```
Array action;
Array precondition;
Array effect;

For x=1 to |R| do
 if (hasSubConcept(R[x])) then
  action[x] = "learn(R[x])";
  if(request(R[x],y)
   precondition[x] = effect(y);
  if(isSubConcept(R[x],y) AND !request(z,R[x]))
   precondition[x] = effect(y);

  effect[x] = "learned(R[x])";

   increase(x)

  action[x] = "learnSubConcept(x)";
   For all subconcept(x) AND !request(z,y)
   then
    precondition[x]=precondition[x] + effect(y)

  if isSubConcept(R[x]) AND !hasSubconcept(R[x])
   action[x] = "learnconcept(x)";

   if !request(y,R[x]) then
     preocondition = effect(isSubConcept(R[x],y))
   if request(y,R[x]) then
     precondition = effect(y)

   effect[x] = "learned(R[x])";
```

appropriate LO for the student. It is Worthwhile to observe that for a concept learning an IPA sequence is necessary. Then each IAA is a sequence of several IPA.

The algorithm 1 describes the IAA generation that takes into account the conceptual map. Thus, given a conceptual map, the IAA are stored in the variables *action*, *precondition* and *effect*, which will be used by the workflow planner. In order to obtain these actions the algorithm uses simple functions such as *hasSubConcept(x)* that returns true if the concept x has a sub-concept; *request(x,y)*, whose result is true if the concept x requests concept y; *isSubConcept(x,y)*, returns true if concept x is a sub-concept of concept y and effect(x) that returns the effect of concept x.

With the data received from Algorithm 1 the Workflow Planner generates a XPDL file containing IAA sequence, defined through Workflow Processes that will be sent to WFMS. The sequence of the IAA sequence defined, the adaptation of the content to the student model is done through IPA.

III. ADAPTATIVE CONTENT SEQUENCE

Adaptation is the DLE capability to make adjustments in the learning process according to the specific characteristics of each student. The main idea is that the system observes the student during the course and dynamically adapts the learning process to each student. However, in order to support the adaptative content sequencing it is necessary to know the preferences, learning style and profile of the student. So, this section presents the student model and the adaptation process of the content sequence for each student.

A. Student Model

A system that intends to be adaptive should collect the student data and determine its profile and learning style. So, it is necessary to observe and register as for example student behavior and prerequisites. Psychological tests aid to determine the student learning style [9]. A student model that contains all of this information was defined by the following fields: (a)Interactivity Level; (b) Semantic Density; (c) Difficulty Level and (d) Learning Styles.

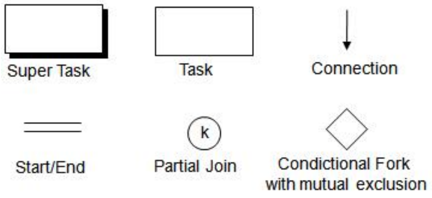

Fig. 3. Representation of the symbols proposed by Casati/Ceri

The Interactivity Level determines how the student interacts with the exposed material. Semantic Density determines which degree of conciseness the material should present. The values for these fields are *very low, low, medium, high and very high*.

The Learning Styles are based on Felder/Silverman model [10]. According to this model, there are four types of student learning styles: (a) Sensitive or Intuitive - The student's preference in relation to the form of material presentation; (b) Active or Reflexive - The way as the student processes the information; (c) Visual or Verbal - The information type that the student prefers to receive and (d) Sequential or Global - The content evolution for the understanding of a concept.

The determination of the student learning style makes use of a psychological test at the moment of the student registration in the course. This evaluation was based on ILS (*Index of Learning Style*) that is an instrument of multiple choice with 44 items for obtaining the preferences in the four scales of the Felder-Silverman Model[10].

Based on the student learning style, is possible to define the strategy that will be adopted with such student. During the course, the instrument that identifies the learning style must be applied to the student several times. According to Terry [11], the preferences modify as the student develops different abilities and the strategies to face new learning processes, especially, in a virtual environment. Besides this, the student model contains the concepts already learned by the student.

B. Workflow Modeling

Workflow modeling and the adaptation will be presented by means the Casati and Ceri model [5]. The Following is a short description of the symbols presented, Figure 3.

- **start and end** - indicates the beginning and end of an Workflow instance ;
- **Partial Join** - indicates that the next task will be implemented after the conclusion of a number k of tasks.
- **Super Task** - is a grouping of several tasks, seeking to reduce the complexity of the workflow;
- **Conditional Fork with mutual exclusion** - indicates that only one activity will be executed after having satisfied determined condition.

C. Content Adaptation based on the Student Model

The adaptation of the content sequence based on the student model is accomplished inside each IAA. At the moment of the execution, an IAA is an argument for the student model.

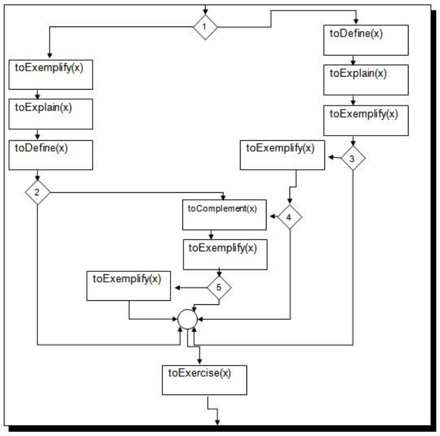

Fig. 4. Super Task learnconcept(x);

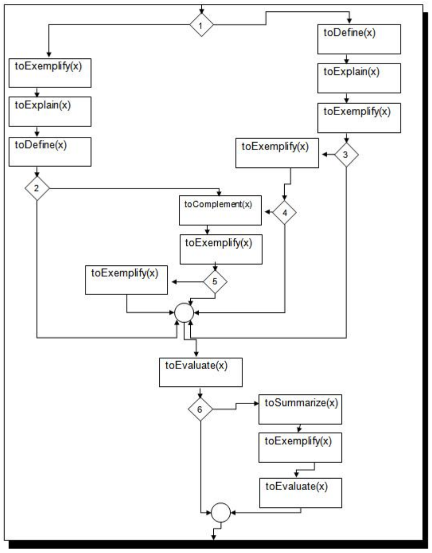

Fig. 5. Super Task learnSubConcept(x);

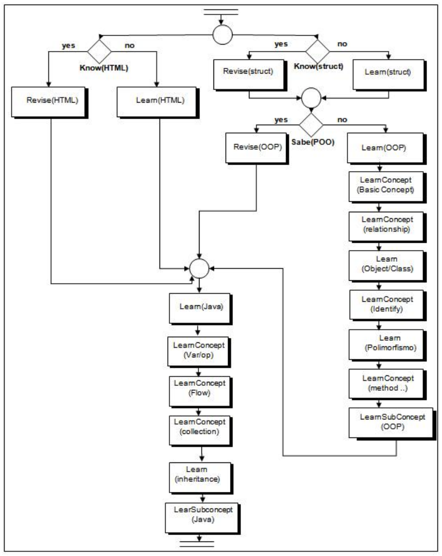

Fig. 6. Workflow for the Java Course.

The IAA *learnconcept ()* and *learnSubConcept()*, as can be seen in Figures 4 and 5, respectively, contains a Conditional Fork 1, which determines the IPA sequence that will be performed by the student. A sequence is designed for each learning style. For instance, a sequence for the sensitive style (related to facts, applications and examples) is given by examples and explanations. The intuitive style (the students like to discover relationships starting from concepts) can have a sequence given by definitions and examples.

The IPA is the main element responsible for the adaptation process of the content sequence to the student model. A LO is selected from the LO repository if it is related to the concept and to the IPA. This selection is done in two steps.

The first step is a search for the fields *taxon* and *purpose* of the *Classification* category of the metadata objects. This category is responsible for storing which concept and what type of instructional activity (to exemplify, to define, and so on) the LO describes, respectively. The results of this search is an LO set. The second one is a search in relation to the *Educational* category. Then, from the LO set of the previous process the LO is selected that presents in the *Interactivity Type*, *Semantic Density* and *Difficulty* fields, characteristics similar to those presented in the student model.

Finally, to determine what type of information will be presented, visual or verbal, based on the student model, the *Learning Resource Type* field of the *Educational* Category to which LO belongs, either to the type visual or verbal is verified.

Also, in the IAA, there are the Conditional Forks 2, 3, 4 and 5. According to the difficulty level, the student is directed to auxiliary IPA, which provides more examples, explanations, complemental materials and exercises.

IV. CASE STUDY

This section describes a case study of workflow modeling generated from the conceptual map of Figure 2.

A. Generated Plan

As has been presented previously, the conceptual map of Figure 2 describes a Java course. The algorithm 1 draws the instructional actions, their pre-conditions and effects. These actions are sent to the workflow planner that generates the content sequencing, in conceptual level, as it has been shown in Figure 6.

In this workflow, it can be observed that the execution of the super task *Learn(Java)* is necessary to execute the super tasks *Learn(HTML)* or *Revise(HTML)* and *Learn(OOP)* or *Revise(OOP)*. The selection between *Learn(HTML)* or *Revise(HTML)* considers if the student has already learned the *HTML* concept. In affirmative case, only a review will be

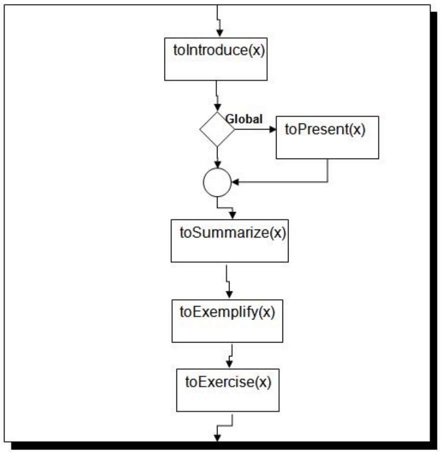

Fig. 7. Super Task Revise(x);

necessary, otherwise, it will be necessary to learn it. The same observations are verified in relation to the concepts *Struct* and *OOP*.

After the execution of the Super Tasks *Learn(HTML)* or *Revise(HTML)* and *Learn(OOP)* or *Revise(OOP)*, the Super Task *Learn(Java)* and the Super Tasks that follow it are executed.

The conditional Forks 2 and 3 of super tasks *learnSubConcept ()* and *learnConcept ()* check the difficulty level. If the student has a difficulty level less or equal to *medium*, the task is executed *toExemplify()*, otherwise, the task *toExercise()* or *toEvaluate()* are executed. This condition aims to aid students with a certain difficulty. The conditional Fork 4 and 5 of the super tasks *learnSubConcept ()* and *learnconcept ()* has similar objectives, where 4 is executed if the student presents a difficulty level less than or equal to *low* and 5 is executed if the student has difficulty level equal to *very low*.

Conditional Fork 6 of the super task *learnSubConcept ()* is executed if the student obtains good results in the evaluation process. Otherwise, more tasks and another evaluation are presented to the student.

The super task *Revise()* is described in Figure 7. As dictated previously, this task is used when the student knows the concept. However, there is the need to review it. In this task, the conditional Fork 1, the student model is verified if it presents Global learning style. If true, the task is executed *toPresent()*, which searches for an LO that makes a global presentation of the concept to be revised.

The super task *learn()* in Figure 8 makes a concept introduction and conditional Fork accomplishes the same condition of conditional Fork of the *Revise()* super task.

V. CONCLUSION AND FUTURE WORK

This work has presented an approach that makes use of AI planning techniques for the generating of Workflow with content sequences to DLE, specially in the sequencing of content based on styles of student learning. It is noticed that with the use of planning for modeling workflow one avoids mistakes, inefficiency and waste of time that occurs in manual modeling. This way, a system that promotes the sequencing of personalized content and management and monitoring of progress is provided.

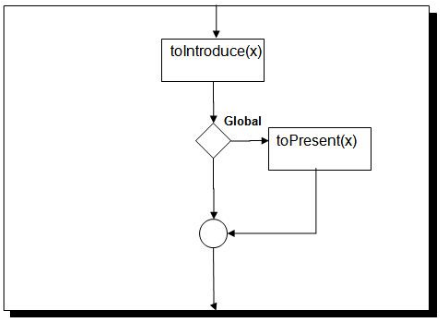

Fig. 8. Super Task Learn(x);

A second stage of the project is the integration of this model to the SimEduc(Multiagent Intelligent System for Distance Education). The architecture of the SimEduc is based in the approach of the intelligent tutoring System and multi-agents Systems. One of the agents that constitutes this system, the pedagogic agent, is responsible for implementing instructional planning , where it is intended to insert the present work. Another future perspective is to allow the student to develop cognitive abilities that are not part of his learning style, through the automatic selection of content of an other learning style, making the student able to learn and work in environments that do not provide the adaptation to his learning style.

REFERENCES

[1] J. P. M. de Oliveira, M. Nicolao, and N. Edelweiss, “Conceptual workflow modelling for remote courses,” in *Workshop Internacional de educação Virtual*, 1998.
[2] C. Ho, “On workflow enabled e-learning services,” in *IEEE International Conference on Advanced Learning Technologies*, 2001.
[3] M. Cesarini, M. Monga, and R. Tedesco, “Carrying on the e-learning process with a workflow management engine,” in *ACM symposium on Applied computing*, 2004.
[4] L. A. M. Pereira, R. N. Melo, F. A. M. Porto, and B. Schulze, “A workflow-based architecture for e-learning in the grid,” in *IEEE International Symposium on Cluster Computing and the Grid*, 2004.
[5] G. R. M. A. SIZILIO, “Técnicas de modelagem de workflow aplicadas à autoria e execução de cursos de ensino à distância,” in *PPGC da UFRGS-Dissertação de Mestrado*, Porto Alegre, RS, Brasil, 2000.
[6] J. Yong, “Internet-based e-learning workflow process,” *International Conference on Computer Supported Cooperative Design*.
[7] F. S. R. Alves, K. F. Guimaraes, and M. A. Fernandes, “Modeling workflow systems with genetic planner and scheduler,” in *ICTAI*. IEEE Computer Society, 2006, pp. 381–388.
[8] “Ieee draft standard for learning object metadada,” *IEEE P1484.12.1-2002*.
[9] P. Brusilovsky and M. T. Maybury, “From adaptive hypermedia to the adaptive web,” *Commun. ACM*, vol. 45, no. 5, 2002.
[10] R. Felder and L. Silverman, “Learning and teaching styles in engineering education,” *Engineering Education*, vol. 78, no. 5, 1988. [Online]. Available: http://www.ncsu.edu/felder-public/Papers/LS-1988.pdf
[11] R. E. Terry, J. N. Harb, P. Hurt, and K. Williamson, *Teaching Through the Cycle - Aplication of Learning Style Theory to Engineering Education at Brighan Young University*. Provo, Utah: BYU Press, 1991.