



PROPOSAL of A LOGICAL MODEL ADAPTATION FOR ADAPTIVE EDUCATIONAL HYPERMEDIA SYSTEMS

AAMMOU Souhaib¹, KHALDI Mohamed², EL KADIRI Kamal Eddine³
^{1,2,3}LIROSA/ Faculty of Sciences,
Abdelmalek Essaadi University,
MORROCO.

Abstract: We present in this paper a logical adaptation model. A model of adaptation must be able to take into consideration the different dynamic aspects of adaptive hypermedia. Such a model is in particular to represent the situation in which the user is located, the way their profile is updated and coping techniques to apply. This model consists of two components: a version of situational logic and rule language.

Keywords: Adaptive hypermedia; Situational logic; Meta-rules languages, Model adaptation.

I. Introduction

Our adaptation model is based on the calculation of first-order predicate. This solution has several advantages. It is declarative and thus more natural to implement the programmatic approaches and less tied to a particular implementation. It is possible to reason about the model, since it is based on logic.

This model consists of two components: a version of the situational logic and rule language. Situational logic, which can represent declaratively situations and how they change based on your actions, provides a basis to represent different aspects of adaptation: representation of the situation in which the user is located, taking into account the changing profile of the latter possibility to perform certain actions. We describe in this article the version of the situational logic that we have chosen for adaptive hypermedia. This logic uses a rule language that we have developed specifically for adaptive hypermedia. This rule language is based on the basic rules and meta-rules. The basic rules are used to describe how to navigate the area, based on the relationships between the various components of the latter. Meta-rules are used to describe what coping strategy, that is to say which set of rules should be selected for a given user. As we will see in this article, this rule-based system using meta-rules can simplify the writing of the rules with respect to existing systems, and can take into account the different types of data at different levels of the model adaptation. We also propose a formal deduction system for describing how to think about the rules of our language. With this deduction system, it is possible to reason about how the rules are taken into account.

II. Problematic

In this section, we present the problems of the current models, and that our model seeks to answer. We present first the problems related to the adaptation of models in general, and in particular the rule language.

A. Model of adaptation

In current systems, adaptation is most often modeled by rules. How these rules are taken into account by the system, the way we reason about its rules and principles underlying adaptation are not explicit. The way the rules involved in the adaptation and how the state of the system is modeled processes are not formally modeled. These aspects are often dependent on a system implementation. They cannot be modified if necessary to take into consideration other forms of accommodation as originally planned, for example. Moreover it is not possible to reason about such systems.

To address this problem, we propose a model that explicitly takes into consideration the different aspects of adaptation, regardless of the implementation that is made. Based on the predicate calculus of the first order, it is possible to reason on this model.

B. Rule's language

The existing rule languages are often based on a single type of rules for describing adaptation. These rules take into account the domain, the user profile and domain knowledge. Syntactically, these elements are found in the premises of the rules. To describe non-trivial adaptation mechanisms, it is necessary to involve numerous assumptions, containing elements of different natures.

The solutions implemented in current systems are unsatisfactory. Moreover, there is no deduction formalisms describing how to interpret the rules, and to change it if necessary. To solve this problem we propose a rule language, using meta-rules.

III. Theoretical approach

A. Situational logic

Situational logic [1] is based on the calculation of first order predicate. It defines a number of predicates, and how to use them to reason. We will see in this section how situational logic is defined and how it works.

Situational logic consists of situations that define the state of the system at a given time that is to say between two actions. The actions change the situations.

1) Situation

A situation is described as the result of an action applied to another situation. The predicate do allow to know the situation obtained by applying an action in a given situation. Thus, $do(a, s)$ defines the position obtained by applying the action to the position s . We can go back to the initial situation. Every situation is then of the form:

$$do(a_n, do(a_{n-1}, \dots, do(a_0, s_0) \dots))$$

where have are actions and where s_0 is the initial situation.

2) fluents

As in reality, only certain aspects of a situation are known. In order to know the value of these aspects in a given situation, there is fluent must be seen as observers of the current situation. For example, in an elevator management problem, a fluent lets you know the current floor. For each fluent, provide:

- The value in the initial situation;
- Its value to a situation of the form $do(a, s)$, ie how it evolves with the actions.

To this end, situational logic defines the predicate holds, for declaring, with rules, the value of a fluent in a given situation. In most cases, the rules defining the fluent possible to calculate the value of the fluent depending on the last action and unprecedented situation this action. Thus, one often defines rules defining holds (fluent, $do(a, s)$) to describe how the situations evolve based on your actions.

3) Actions

Actions allow you to change situations. Fluent cannot possibly change value after an action, and only if it has an influence on the appearance of the situation described by the fluent.

Actions are categorized along two orthogonal axes. In a given situation, there are two types of shares situational logic. The ability to perform an action depends on the current situation. To this end, the *poss* predicate is defined. It allows to describe, using rules, what conditions are necessary for action to be possible. The possibility of rules of the form:

$$premisses \Rightarrow poss(action, situation)$$

We select an action from the possible actions. The ability to perform an action can be provided by one or more rules, described as Horn clauses. [2]

The composite action describe the sequences of actions or equity disjunctions (random selection), for example. There are more details on the composite action in [3].

B. Planning Problem

To choose a formalism to represent merits adaptation in adaptive hypermedia systems, we chose to study various planning formalisms. Indeed, provide content tailored in such a system returns to predict, as the actions taken by the user, the succession of documents that can or must submit. It therefore seems natural for us to concern ourselves with approaches used in the field of planning to formalize the problem of adaptation in our field. The scale of data to be processed is nevertheless very different in the two cases: in adaptive hypermedia, the amount of data to be processed - all the components of the field and a user's attributes - is much lower than in planning issues. Therefore, we are not particularly interested treaties optimization problems in planning systems.

1) Formalisme STRIPS

In this section, we give some basic elements to model a planning problem using the STRIPS formalism (STanford Research Institute Problem Solver).

A planning problem consists of an initial situation (the one from which it is desired to solve the problem), a final situation (the one we want to happen) and intermediate situations. Each situation has different facts or atoms. To go from one situation to another, it performs operations. An operation changes the situation by changing the atoms of the situation: some atoms cannot be satisfied, and others can become.

It is these considerations what built the STRIPS model. In this formalism, a planning problem is a tuple (A, O, I, G) . A represent a group of atoms, O is a set of operators, I is the initial situation, G is the final situation. I and G are subsets of A .

For each operator, we define three subsets of A :

1. $Prec(op)$, which defines the preconditions op atoms i.e. those that should be part of the current situation in order to apply it op ;
2. $Add(op)$, which defines the atoms that op adds to the current situation
3. $Del(op)$, which defines the atoms that op withdraws to the current situation.

From this, it is possible to construct a graph of all possible states. Planners aim to find the shortest path from the initial situation to the final situation.

2) Heuristics planning model

The heuristics planning model proposed in [4] is based on a modified version of situational logic. We will detail here the differences between the basis of situational logic, and one used for planning.

The selection of actions to be applied in the version of the situational logic proposed in this article is by heuristic rules, which determine the value of three predicates:

1. *Good* predicate that characterizes the optimal actions. So $Good(a, s)$ is true if the action is good in situation s , ie if it leads to a state approaching the target.
2. The predicate *Bad* characterize the actions that cannot be part of an optimal plan. Thus, we will never make the action a in situation s if $Bad(a, s)$ is true.
3. The predicate *Better* provides a partial order for actions that are neither *Good* class nor *Bad* class. $Better(a, b, s)$ is true if the action is better than action b in situation s . Note that we can easily extend this partial order to the actions of *Good* classes. If two actions are good, it may be interesting to determine which one is preferable.

These rules are used as follows: s in a given situation, we want a share as $Good(a, s)$ and applied. If there are several, we chose a "random". If we find no action as $Good(a, s)$, we seek from those such as $\neg Bad(a, s)$ the best (or the best) in the sense of partial order *Better*. If all the shares are such that $Bad(a, s)$, then returns to the previous situation and selecting another action.

IV. Meta-rules

A. Overview

In this part, we are particularly interested in systems using meta-rules. Meta-rules allow to stratify the design of a rule-based system on multiple levels, avoiding each of these levels becomes difficult to implement and maintain. Meta-rules are a possible solution to our problem.

Meta-rules are used in different kinds of systems. They are especially widely used in expert systems. They are also found in the analysis of natural language systems, control systems, database, etc.

We focus on the system of Jagadish [5], which we thought was particularly interesting because it offers a simple way to describe the dependencies between rules and allows for checks on the correction of meta-rules, which we will base to create a meta-rule system specific to adaptive hypermedia.

B. The meta-rules Jagadish system

It seemed particularly interesting to detail the meta-rule system introduced by the authors of [5]. Indeed, it is a formal system meta-rules on which it is possible to perform a number of checks. In particular it is possible to know whether a rule will never be executed, if the system is deterministic or if the order obtained for the rules to be executed is unique. All of these calculations is feasible in polynomial time depending on the size of the sets of rules and meta-rules. In this section, we describe in detail the system initially for automatic updating databases.

A system of rules is given by $S = \langle V, M \rangle$ where V is a set of rules and M is a set of meta-rules.

1) Definitions

A rule (i.e. an element of V) consists of a trigger or trigger, premise and a conclusion. Each trigger is a set of rules, the set V presented above.

A rule is said to be triggered if all its premises are true. One note I the subset of V triggerable rules in a given state of the system.

The subset of rules that are actually executed after selection and scheduling of meta-rules, is denoted O .

We have the following trivial relation: $O \subseteq I \subseteq V$.

2) Types of meta-rules

There are four types of meta-rules throughout M . These meta-rules allow control of interactions and enforcement of the rules of V . Here are four types, with $A, B \in V$:

- Meta-rules positive requirement. Let $A \supset B$. If A is executed, B must be too. We say that " A requires B ".
- Meta-rules mutual exclusion. We note $\overline{AB}$. The rules A and B cannot be simultaneously selected for execution. We use notation $\overline{A}$ rather than $\overline{AA}$ in the case of a mutually exclusive rule itself, i.e., a dead rule.
- Meta-rules preference. We note $A > B$. If A and B are both releasable and if they cannot be executed together, then we must choose A .
- Meta-rules scheduling. We note $A < B$. If A and B are both selected for execution, execute A before B .

3) Axioms Inference

The following axioms are used to account for the possible inferences at the meta-rules. They form a complete and consistent whole. They reveal whether a meta-rule γ can be inferred from a set of meta-rules Γ .

Here these axioms: $A1 : \overline{AB} \vdash \overline{BA}$

- $A2 : (A \supset B \wedge \overline{BC}) \vdash \overline{AC}$
- $A3 : \overline{A} \vdash \overline{AB}$
- $A4 : \vdash A \supset A$
- $A5 : (A \supset B \wedge B \supset C) \vdash A \supset C$
- $A6 : \overline{A} \vdash A \supset B$
- $A7 : \vdash A > A$

- A8 : $(A \supset B \wedge C > B) \vdash C > A$
- A9 : $(A < B_1 \wedge B_2 \dots B_k < B) \wedge A \supset X \wedge B \supset Y \vdash A < B$ où $X \cup Y = \{B_1, B_2, \dots, B_k\}$
- A10 : $(B_1 < B_1 \wedge B_2 \dots B_k < B_1) \wedge A \supset X \wedge B \supset Y \vdash \overline{AB}$ où $X \cup Y = \{B_1, B_2, \dots, B_k\}$
- A11 : $\overline{AB} \vdash A < B$

The implication problem for meta-rules, that is to say, the problem is whether or not we can deduce a meta-rule from M and axioms A1 to A11, is decidable in polynomial time.

4) Execution sequence

Once all the necessary checks carried out, we need to calculate, for I gave, the ordered set O of the rules to run. The algorithm is the following:

1. Initialize O as equal to I;
2. For any pair of rules A, B such that $A \in O$ and $B \notin O$ if $M \vdash (A \supset B)$ then remove A O and remove any rule $C \in O$ as $M \vdash (A > C)$;
3. Identify the set D of minimum rules of mutual exclusion;
4. The system being assumed deterministic, for each meta-rule AB D, it is known if A or B is preferable if the opposite is true. Suppose $M \vdash (A > B)$:
 - a. Delete B from O;
 - b. Delete any rule $C \in O$ such that either $M \vdash (C \supset B)$ or $M \vdash (B > C)$.
5. Establish a total order on O using $A < B$ type of meta-rules.

V. Construction of our adaptation model

In the previous section, we described a number of techniques that we used to build the model of adaptation that we present in this section. First, we describe our model-based situational logic, which is based on certain planning principles described in the previous section.

A. Situational logic-based model's

We will detail here the situational logic model that we have established for adaptive hypermedia [6, 7, 8, 9].

The operation that we give to the system is:

```

While (the target user is not reached)
{
  // Let i be the highest degree of desirability reached by at least one share in the current situation.
  If (i = 0)
    No action possible;
  else
    It offers the user to select from all the actions of degree i;
  If (the action is performed)
    User data is updated - ie fluent are changed;
}

```

We have therefore developed a model based on situational logic, which allows to represent coping mechanisms for adaptive hypermedia. This version of situational logic provides a logical framework to structure the data required for the adaptation: fluent, actions, rules. The most important feature of these data is the specific rule system that we have defined above situational logic, and we will describe in the next section.

B. A rule language for adaptive hypermedia

We now present in detail the formal rule language that we propose to address the problem. This language is based on the Jagadish system. It takes into account all the characteristics described in previous sections. We will first define the whole system and then show how the properties in the Jagadish verifiable system can be in ours.

1) General definition

System of rules called a tuple $S = \langle V, F, M, \delta \rangle$ where:

- V is a set of route policy rules;
- F is a set of trigger criteria;
- M is a set of meta-rules;
- δ is a function of association between the rules V and F. criteria

We will now detail the nature of each element of the quadruplet S.

2) Set of rules of strategy courses

V is a set of rules of the form:

$$\forall X \in E \left(\bigwedge_{i \in I} P_i(X) \Rightarrow \text{Deg}(\text{Action}(X), d) \right)$$

Notations have the following meanings:

- I is finite set of indices.
- R is a rule identifier.
- d is the degree of desirability of action.

E is a set of components in the domain of one of the following forms:

- E is of the form $\{X / \text{Type}(X, _)\}$, that is to say, it represents all the components of a given type.

- E is of the form $\{X / XR_ \}$ or $\{X / _RX\}$, that is to say, it represents all the constituents in relation to a given component.
- E is equal to the constant Cst, which is defined as all of the components of the field.

Pi (X), the premise of a rule is of the form:

- $P_i(X) = XR_ \text{ or } _RX$, that is to say, the premise is that X is in contact with a component.
- $P_i(X) = XR^*_\text{ or } _R^*X$, that is to say, the premise is that X is in contact with a component transitively.
- $P_i(X) = XR^n_\text{ or } _R^nX$, that is to say, the premise is that X is transitively connected with a component, and that the length of the path connecting transitive X and the component is less than n.
- $P_i(X) = \text{metadata}(X, \text{property}, \text{value})$, ie that the premise is testing a value of metadata associated with the component X.
- $P_i(X) = \forall Y \in F(\bigwedge_{j \in J} P_j(X, Y))$, where F is of the form E, where J is of the form I and where $P_j(X, Y)$ has the form of $P_i(X)$, but depending on X and Y. Thus, a premise can itself be a conjunction universally quantified premises.
- $P_i(X) = \exists Y \in F(\bigwedge_{j \in J} P_j(X, Y))$, where F is of the form E, where J is of the form I, and where $P_j(X, Y)$ has the form of $P_i(X)$, depending on X and Y. Thus, an assumption can be itself an existentially quantified conjunction of premises.

3) Set of trigger criteria

As we have explained, it is not feasible in the context of adaptive hypermedia [10], whether all the premises of a path rule are true in a given situation. Indeed, in the same situation, every rule is applied to a number of potential actions. We decided to separate the rules explicitly trigger criteria, to solve this problem. These trigger criteria should involve only the user model. In any case, they must involve the field of materials or knowledge that the user of the domain.

Thus, we separate the two aspects "domain" and "user" as we wanted. Rules V are exclusively assigned to the choice of the route depending on the positioning of the user in this way. F criteria will preset rules batches triggerable for a given type of user.

C. Inference Formulas

To make inferences about our system of set meta-rules, we have transformed each meta-rule set-in single meta-rule, and the algorithms described are used in [5].

We now describe an inference system on the set-meta-rules. We leave the axioms of the system and Jagadish transpose the same written inference wording changing the ground rules by rule sets.

We will "translate" the axioms A1 to A11 of Jagadish system, replacing the rules by rule sets. Throughout this section, A, B and C are sets of rules.

- F1 : $\overline{AB} \vdash \overline{BA}$
- F2 : $(A \supset B \wedge \overline{BC}) \vdash \overline{AC}$
- F3 : $\overline{A} \vdash \overline{AB}$
- F4 : $\vdash A \supset A$
- F5 : $(A \supset B \wedge B \supset C) \vdash A \supset C$
- F6 : $\overline{A} \vdash A \supset B$
- F7 : $\vdash A > A$
- F8 : $(A \supset B \wedge C > B) \vdash C > A$
- F9 : $(A < B_1 \wedge B_2 \dots B_k < B) \wedge A \supset X \wedge B \supset Y \vdash A < B \text{ où } X \cup Y = \{B_1, B_2, \dots, B_k\}$
- F10 : $(B_1 < B_1 \wedge B_2 \dots B_k < B_1) \wedge A \supset X \wedge B \supset Y \vdash \overline{AB} \text{ où } X \cup Y = \{B_1, B_2, \dots, B_k\}$
- F11 : $\overline{AB} \vdash A < B$

However F9 pose problem. Indeed, we F4, i.e. we do not always $A \supset A$, so if $A < A$, we cannot prove A. It is therefore proposed to add the following axiom:

- F12 : $(A < B \wedge B < A) \vdash \overline{AB}$

This set proves that we can make some inferences about the set-meta-rules.

The set of formulas allows us to accelerate the calculation of meta-rules can be inferred. We propose the following algorithm to accelerate Jagadish method of using the writing of set and formulas F1 to F12:

While (stable state == false)

```
{
    Applying F1 to F12 meta-set-rules; // step 1
    Meta-rules is deduced from the whole of set inferred the corresponding
    set of individual meta-rules; // step 2
    Applying A1 to A11 Inferred individual meta-rules; // step 3
}
```

In this way, we arrive at the same result as directly through individual meta-rules. The calculation is faster because the set-meta-rules are much less numerous than the corresponding individual meta-rules, and inferences

are the first step in polynomial time based on the number of set meta-rules. The second step is necessary in any case if transcribed first meta-set-rules as individual meta-rules. The last step is greatly accelerated, since much of the meta-rules inferable will already have been inferred in step 1.

VI. Conclusion

In this paper, we presented an adaptive system using meta-rules, specifically designed for adaptive hypermedia. This system allows to adapt the strategy itself (in addition to the calculated route) depending on the user, without making the basic rules difficult to design and maintain. It is important to note that the rules established system in this paper can generate real strategies course in itself, different depending on the profile, abilities, and preferences of the user. It remains to show that this system allows for a number of desirable audits, so as to facilitate the task of adaptive hypermedia designers.

VI. References

- [1] Raymond Reiter. Proving properties of state in the situation calculus. *Artificial Intelligence*, 64(2) :337-351, 1993.
- [2] Alfred Horn (1951), "On sentences which are true of direct unions of algebras", *Journal of Symbolic Logic*, 16, 14–21.
- [3] Hector J. Levesque et al. Golog : A logic programming language for dynamic domains. *Logicbased artificial intelligence*, pages 257 – 279, 2000.
- [4] Josefina Sierra-Santibañez. Heuristic planning : A declarative approach based on strategies for action selection. *Artificial Intelligence*, 153 :307–337, 2004.
- [5] H. V. Jagadish, Alberto O. Mendelzon, and Inderpal Singh Mumick. Managing conflicts between rules. *J. Comput. Syst. Sci.*, 58(1):13–28, 1999.
- [6] C. Jacquot, Y. Bourda, and F. Popineau. Geahs : A generic educational adaptive hypermedia system. *World Conference on Educational Multimedia, Hypermedia & Telecommunication (ED-Media 2004)*, pages 571–578, 2004.
- [7] Cédric Jacquot, Yolaine Bourda, and Fabrice Popineau. Geahs : A generic educational adaptive hypermedia system based on situation calculus. In Bra and Nejd [10], pages 413–416.
- [8] Cédric Jacquot, Yolaine Bourda, and Fabrice Popineau. Reusability in geahs. In *ICWE Workshops*, pages 199–209, 2004.
- [9] Cédric Jacquot, Yolaine Bourda, Fabrice Popineau, Alexandre Delteil, and Chantal Reynaud. Glam : A generic layered adaptation model for adaptive hypermedia systems. In Wade et al. [16], pages 131–140.
- [10] Souhaib, A., Mohamed, K., Eddine, E., & Ahmed, I. (2010). Adaptive Hypermedia Systems for E-Learning. *International Journal of Emerging Technologies in Learning (iJET)*, 5(2010).