Concept Relationship Types for AHA! 2.0

Paul De Bra, Ad Aerts, Brendan Rousseau
Department of Computer Science
Eindhoven University of Technology (TU/e)
PO Box 513, Eindhoven, The Netherlands
{debra,wsinatma,broussea}@win.tue.nl

Abstract: In adaptive Web-based applications such as on-line course material, virtual museums and shopping sites, there are different types of relationships between objects, pages or high-level concepts. They indicate that some concept is a prerequisite for another concept, that a page or object exemplifies a concept, that a user reading a page gains knowledge of or shows interest in a concept, etc. In AHA! version 2.0 (De Bra et al, 2002) such relationships are expressed at a very low level through adaptation rules. In this paper we show how to define higher level *concept relationship types* (such as prerequisites) and how to automatically translate instances of these types, called *concept relationships* to adaptation rules for AHA! This work is part of the Adaptive Hypermedia for All project of the NLnet Foundation.

Introduction

When the AHA! system was first built and used for an on-line course (on the subject of hypertext and hypermedia) the relationships between pages and concepts were simple: reading pages contributed knowledge about concepts, and knowledge about certain concepts was a prerequisite for reading other pages (or for the inclusion of fragments with additional explanations). See (De Bra & Calvi, 1998) for a description of an early version. AHA! version 1.0 introduced a bit, and version 2.0 a lot more possibilities for describing the interaction between pages and concepts. In (De Bra et al, 2002) we describe the powerful adaptation rule language used in AHA! 2.0 that lets authors specify how a rich user model should be updated using *condition-action rules*. Early comments we received on these new possibilities indicated that AHA! 2.0 offers what can best be described as an *assembly language for adaptation*. It became clear that higher-level descriptions of the relationships between concepts were necessary. In this paper we describe how the notion of *concept relationship types*, inherited from the AHAM reference model (De Bra et al, 1999), can be incorporated into the AHA! system. We first describe how to define a concept relationship type, using an XML notation that is similar to that of AHA! 2.0. We show how a simple drawing of concept relationship structures (possibly using different relationship types) can be translated automatically to the combined domain/adaptation model of AHA! 2.0. This idea has led to the development of a new authoring tool that aims to bring back to AHA! what was lost through the introduction of the *condition-action rule language*: ease of use for authors.

Defining and Using Concept Relationship Types

Concept relationships are a common notion in adaptive Web-based applications. They can be used to represent *prerequisites* in an adaptive course and to propagate *knowledge* from pages to sections and from sections to chapters. This is essentially how systems like Interbook work (Brusilovsky et al, 1998). In a museum there may be *exemplifies* relationships between paintings and a painting style or a *painted by* relationship between paintings and painters. In an on-line shop products may be *competing with* other products or may be *complementing* other products. The overview papers (Brusilovsky, 1996) and (Brusilovsky, 2001) show the ranges of applications in which adaptive hypermedia techniques are used. The AHA! system is not aimed at one specific application area, but tries to offer general-purpose functionality. It uses a combined domain/adaptation model to indicate how concepts relate to each other and how the system should update the *user model*. In (De Bra et al, 2002) the rule language of AHA! version 2.0 is explained in detail. A drawback of this approach is that *concept relationships* are described through their effect on the *user model* only. Not only are the adaptation rules difficult to write, in applications where different types of relationships play a role it is also difficult for authors to recognize which relationships are expressed by which adaptation rules. We therefore propose a new

authoring tool that uses *concept relationship types* and that provides automatic translation from *concept relationships* of a defined type to AHA! adaptation rules. In the terminology of the AHAM reference model (De Bra et al, 1999) this means we create an (at least partial) separation between the *domain model* and the *adaptation model*.
We illustrate the use of concept relationship types through the examples of *prerequisite* and *knowledge propagation relationships*, as used in educational applications. The following figures could represent a few concepts in an imaginary course on Web concepts, how they contribute knowledge to the entire course topic, and how they "depend" on each other:

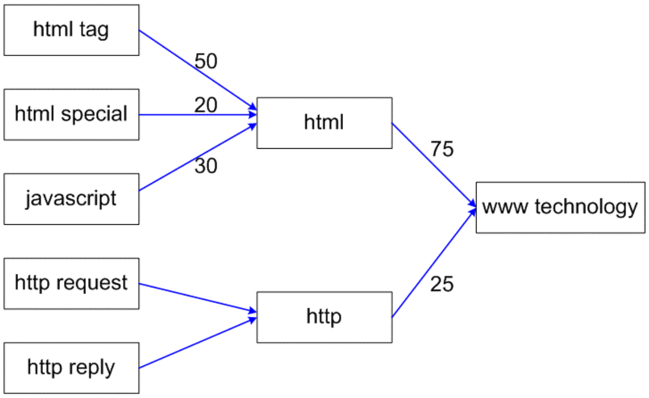

Figure 1: knowledge (update) structure in a course.

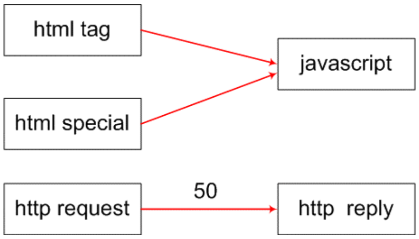

Figure 2: prerequisite relationships in a course.

From the graphs we observe the following properties of these concept relationship types:

- The relationships are all *binary*. We see that we can use two or more relationships with the same (source or) destination, to indicate that several pages contribute knowledge to a concept or that several pages are a prerequisite for a single concept. We shall see below how this affects the generation of the adaptation rules.
- The graphs are both *acyclic*. This may not be true for all imaginable relationship types. An authoring tool can verify whether the property holds, if needed.
- Relationships have an optional value associated with them. For the knowledge this indicates how much knowledge the source concept contributes to the destination. (In terms of a concept *hierarchy* this would be contribution of *child* concepts to *parent* concepts.) For the prerequisites it indicates how much knowledge of the source concept is required before the destination concept becomes "suitable".

In AHA! entering concept relationships becomes as simple as drawing a (labeled) graph. (We do not pretend that *designing* a proper structure of concept relationships is simple.) The new AHA! concept relationship authoring tool can show the graph for each concept relationship type separately or can overlay the graphs using different colors or line styles. Below we show two examples: the definition of *knowledge propagation* and the definition of the *prerequisite relationship type*.

In the first example we show (in a simplified way) how knowledge is propagated. (Note that another concept relationship type is needed to make a page access generate knowledge about a page. The example below only shows how knowledge is transferred from one concept to another.)

```
<concept_relation_type>
  <name>knowledge propagation</name>
  <color>blue</color> <style>full</style> <properties acyclic=”true”/>
  <listitems>
    <generatelistItem location=”source.knowledge” isPropagating=”true”>
      <requirement>true</requirement>
      <trueActions> <action combination=”DIV_S”>
        <concept>destination</concept>
        <attribute>knowledge</attribute>
        <expr>destination.knowledge + var:DIVIDE * _source.knowledge</expr>
      </action> </trueActions>
    </generatelistItem>
  </listitems>
</concept_relation_type>
```

Using this concept relationship type definition the authoring tool can translate the knowledge update edge between “javascript” and “html” (in Figure 1) to the following AHA! code:

```
<concept>
  <name>javascript</name>
  ...
  <attribute name=”knowledge” type=”true” isPersistent=”true” System=”false”>
  ...
  <generatelistItem isPropagating=”true”>
    <requirement>true</requirement>
    <trueActions> <action>
      <concept>html</concept>
      <attribute>knowledge</attribute>
      <expr>html.knowledge + 0.3 * _javascript.knowledge</expr>
    </action> </trueActions>
    </generatelistItem>
  </attribute>
  ...
</concept >
```

This example shows the following elements in the translation:

- The generated adaptation rule belongs to the “knowledge” attribute of concept “javascript” (the source), so whenever this attribute value changes the rule is triggered. (This rule has no condition: the “requirement” is simply “true”. The condition is translated as well, but here it is just trivial.)
- The action of this rule updates the “knowedge” attribute of the concept “html” (the destination). The *change* in the “knowledge” value of “javascript” is used. This is indicated by the prefix “_”.
- When several knowledge propagation relationships have the same destination the fraction of the knowledge value change that is actually propagated is calculated by dividing 1.0 by the number of sources. However, in the example of “javascript” and “html” the divider that would be calculated by default is overridden by the label “30” in the graph. (Allowed because of “`var:DIVIDE`”.) This results in the division factor 0.3. (For “http request” and “http reply” the factor is 0.5.) The term “`DIV_S`” indicates division by *source* of the relationship, “`DIV_D`” would mean division by *destination*.

```
<concept_relation_type>
  <name>prerequisite</name>
  <color>red</color> <style>full</style> <properties acyclic=”true”/>
  <listitems>
    <generatelistItem location=”destination.suitability”>
      <requirement>true</requirement>
      <trueActions> <action combination=”AND”>
        <concept>destination</concept>
        <attribute>suitability</attribute>
        <expr>source.knowledge > var:70</expr>
      </action> </trueActions>
    </generatelistItem>
  </listitems>
</concept_relation_type>
```

The prerequisites “html tag” and “html special” of the concept “javascript” (in Figure 2) result in the following piece of AHA! code:

```
<concept>
  <name>javascript</name>
  ...
  <requirement>
    html_tag.knowledge > 70 && html_special > 70
  </requirement>
  ...
</concept>
```

This example shows the following elements in the translation:

- The prerequisites contribute to the “destination” concept (in this case “javascript”).
- The “suitability” attribute mentioned in the rule is actually a pseudo-attribute. Using it in a concept relationship type means that it is the “requirement” of the whole concept that will be created.
- The “source.knowledge > var:70” expression means that the knowledge value of the source concept (html_tag or html_special) is compared with 70 or with a number given as label in the graph. (The real XML representation uses &gt; instead of > because of XML restrictions.)
- The “AND” combination of the action means that the expressions for different prerequisites are combined using the logical *and* operator (represented by && here). It means that all prerequisites for a concept must be satisfied in order for it to be considered suitable. (For other types of relationships the combination may also be “OR”, meaning that at least one of the conditions must be satisfied. Combinations of requirements from relationships of different types, having the same destination, are always combined with “AND” (and thus not explicitly written in the definitions).

Conclusions and Future Work

In this paper we have shown how drawings of concept relationships can be translated to AHA! 2.0 concept and adaptation structures. In order to do so an AHA!-like syntax was developed to define concept relationship types and their translation to adaptation rules.

The translation of concept relationships to AHA! is fairly straightforward, even in the presence of relationships of several different types in the same application. A prototype interface enables authors to create concept relationships and have them translated to AHA!. It contains the definition of a number of commonly used relationship types. In the future we will also develop an interface for defining new types and add the ability to author adaptation rules at the presented high level and the low level of AHA! 2.0 at the same time.

References

Brusilovsky, P. (1996). *Methods and Techniques of Adaptive Hypermedia*, User Modeling and User-Adapted Interaction (6), pp. 87-129, Kluwer academic publishers.

Brusilovsky, P. (2001). *Adaptive hypermedia*. User Modeling and User Adapted Interaction, 11, (1/2) pp. 87-110.

Brusilovsky, P., Eklund, J., Schwarz, E. (1998). *Web-based Education for All: A Tool for Developing Adaptive Courseware*, Computer Networks and ISDN Systems (Seventh International World Wide Web Conference), vol. 30, 1-7, pp. 291-300.

De Bra, P. and Calvi, L. (1998). *AHA! An open Adaptive Hypermedia Architecture*, The New Review of Hypermedia and Multimedia, 4, pp. 115-139, Taylor Graham Publishers.

De Bra, P., Aerts, A., Smits, D. and Stash, N. (2002). *AHA! Version 2.0, More Adaptation Flexibility for Authors*, Proceedings of the AACE ELearn'2002 conference, Montreal, Canada.

De Bra, P., Houben, G.J., Wu, H. (1999). *AHAM: A Dexter-based Reference Model for Adaptive Hypermedia*, Proceedings of the ACM Conference on Hypertext and Hypermedia, pp. 147-156, Darmstadt, Germany.