Understanding Adaptive Hypermedia: An Architecture for Personalisation and Adaptivity

James Ohene-Djan
Matt Gorle
Department of Computing
Goldsmiths College
University of London
SE14 6NW, UK
j.djan;m.gorle@gold.ac.uk

Christopher P Bailey
Gary B Wills
Hugh C Davis
Department of Electronics and
Computer Science
University of Southampton
SO17 1BJ, UK
cpb;gbw;hcd@ecs.soton.ac.uk

ABSTRACT

The predominant approach taken by the adaptive hypermedia (AH) community to implementing AH systems may be characterised as application-orientated and technology-driven. This has, in part, resulted in relatively low levels of interchangeability and interoperability between AH systems. To address this situation recent AH research has begun to focus on understanding the mechanisms and structures underpinning AH systems. In particular, several formal models have been proposed which address these issues, the prominent examples being Adaptive Hypermedia Application Model, the Fundamental Open Hypermedia Model and the Goldsmiths Adaptive Hypermedia Model.

In this paper we propose a component-based architecture for AH within which these approaches to modelling are compared and understood. Using this architecture we show that although each model uses different representations there are important underlying commonalities. It is envisaged that through an understanding of these commonalities, it may be possible for the AH community to devise a standardised "plug 'n' play" architecture for the development of future AH systems.

Categories and Subject Descriptors

H.5.4 [**Information Interfaces and structures Presentation**]: Hypertext/Hypermedia—*Architectures, Theory*

General Terms

Design, Human Factors, Modelling

Keywords

adaptation, adaptive hypermedia modelling, component technology, construct, design, hypertext, hypermedia, user model

1. INTRODUCTION

Adaptive Hypermedia (AH) systems use knowledge provided by (or captured about) specific users to tailor the information and links presented to individual users. By applying the knowledge they accumulate, adaptive features are made available that support users in navigation and information acquisition. Such support may include limiting options for traversal to information units, tailoring content and presentation of information units, suggesting relevant links to follow and providing additional information on links and information units.

Early AH systems were devised to address a particular application area or information domain and were implemented with whatever technologies were available, guided by the intuition of their developers. With the advent of engineering-oriented approaches [18], researchers began to formalise the process of building AH systems. Brusilovsky has provided an account of early AH research [5]. Surveys of more recent AH concepts and the research that has given rise to them can be found in [6].

Current research has started to focus on the mechanisms and structures present in AH systems. In particular, several formal models, emphasising the importance of hypermedia structures and their manipulation, have been proposed. Qualifying representative examples are the Adaptive Hypermedia Application Model (AHAM) [10], the Fundamental Open Hypermedia Model (FOHM) [1] and the Goldsmiths Adaptive Hypermedia Model (GAHM) [23].

In this paper we propose a general, component based architecture for AH within which the AHAM, FOHM and GAHM approaches to modelling AH are compared. Each model takes a structure-oriented approach, in which emphasis is placed on the structures used to specify hyperdocuments and their meta-data rather than the content they contain. We illustrate that although each model uses different representations of hypermedia structures and has differing approaches to manipulating these structures, there are important underlying commonalities in terms of the components found within AH systems and the interactions between them.

For the purposes of this paper, we make a distinction between *Personalisation* and *Adaptation*. Any action that alters the structure of a hyperpage, or one of its' component parts, is referred to as a *tailoring action*. *Personalisation* actions are user-initiated tailoring actions. *Adaptive* actions are system-initiated personalisation actions.

The remainder of this paper is structured as follows. Subsection 1.1 outlines the motivating factors behind the research reported. Section 2 presents an architecture for general AH systems within which the key components found within AH systems are delineated and their roles described. Using this architecture as a template, AHAM, FOHM and GAHM are described in Sections 3, 4 and 5. Section 6 presents a discussion of the opportunities afforded by the proposed architecture. Section 7 compares our results with those of others and draws conclusions.

1.1 Motivation

Although AH research has delivered a variety of systems, [9, 7, 12, 13, 19, 22, 26] there is still no consensus as to the components that should comprise such systems. The research reported in this paper is motivated by the view that such a consensus could be achieved through a generalisation of the components within AH systems and the interactions between them. With such a consensus we believe that it is possible to facilitate a systematic investigation of the space of possibilities for personalisation and adaptation (P&A) actions.

The technology-driven approach towards the development of AH systems that has prevailed makes a principled testing of the benefits of P&A actions difficult. In many cases it is uncertain whether what is being tailored is a unique, distinctive, property of hypermedia-based interaction or instead simply arises as a consequence of coupling user-interface, database and link management components in a particular manner.

Therefore a further motivating factor of our research is how to devise an architecture, at a suitable level of abstraction, that can represent the functionality that is unique to AH systems. Such an architecture should clearly delineate the discrete groupings of functionality found within AH systems, so that they may be viewed as interchangeable and, possibly, interoperable components of AH systems.

In the context of this paper, we view *personalisation* as the process of handing over to the user the ability to take actions to tailor hyperpages, thereby overriding, in principle, each aspect of a hyperpages' content and presentation. *Adaptivity* is viewed as the process of allowing the system to take the initiative in tailoring actions in the light of the systems' inference of a users' information goals and history. We argue that system-initiated tailoring (adaptivity) is therefore, in principle, as expressive as user-initiated tailoring and requires no technologies other than those involved in user-modelling and in decision-making from a user-model.

With respect to the view above, a final motivating factor is to devise an architecture which accommodates personalisation yet treating the components required to perform adaptation, (i.e., a user-model and decision-making algorithm) as "plug 'n' play" black box components. We view this as beneficial, as it would enable appropriate user-models and decision-making algorithms for particular application areas to be included as and when required.

To address these motivating factors, in Section 2, we detail a general, open architecture for personalisation and adaptation. This architecture reflects the predominant approaches to the design of AH systems and aims to illustrate the interactions that take place between their components.

2. THE ARCHITECTURE

A general, open architecture for hypermedia systems, of the kind depicted in Figure 1[1], is assumed. This architecture reflects the predominant approach to the design of hypermedia systems, whether web-based or otherwise. It is assumed that a core of AH functionality is a client technology loosely coupled to one or more user interface servers (UISs) and database servers (DBSs). An example UIS is a web-browser, and an example DBS is any Database Management System that supports client/server architectures (e.g. Oracle, PostGreSQL).

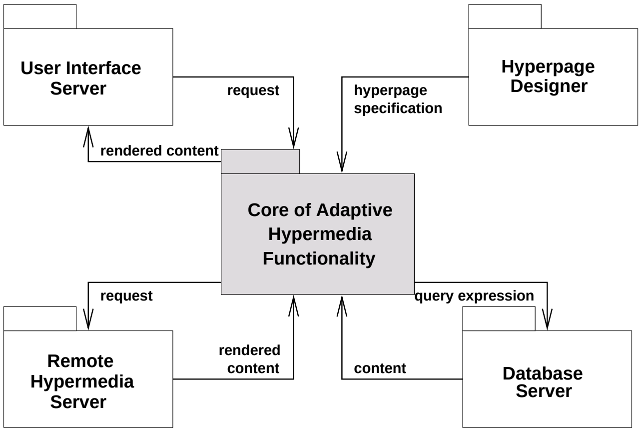

Figure 1: A General, Open Architecture for Hypermedia Systems

UISs capture requests for hyperpages. These are channelled into the core of AH functionality. If a request is for a local hyperpage (i.e., one known to the core), the core responds by composing a rendering expression that can be rendered by UISs, possibly after querying one or more DBSs to fetch some or all of the content required by the requested page. It is assumed that the AH system may rely on other hypermedia systems, in the role of a client/server, in a similar manner to the server-browser model in the WWW. In this case the client AH system plays no active role, other than to forward requests received from the UIS to the remote server and return responses from the latter to the former.

Implicit in Figure 1 is the assumption that P&A actions in AH systems are independent of any adaptive actions that might be provided by UIS and DBS components of a hypermedia system. This core of AH functionality, the shaded area in Figure 1, is the component that performs the tailoring and is responsible for what users experience as hypermedia-based information retrieval.

---

[1] This and all further diagrams use classical UML notation.

In the following subsection, we outline our view of these structures. Following this, we describe the components within the core of AH functionality, namely the *user-model* and *composer & tailoring engine*. Figure 2 depicts the architecture of the core of AH functionality.

2.1 Hypermedia Structures

The simplest form of structure is a single, flat formal text, or hyperpage. However, to tailor the information contained within such a page, it is necessary to subdivide it into information units.

A hyperpage can be viewed as a sequence of information units (containing, for example, text, numbers, graphics or video), or nodes. These unit have associated with them, a set of references to other units. Each information unit is comprised of a specification of its' content and a specification of how to present (or render) its' content. A specification of content may be the content itself or a reference to the content in the form of query expressions that can be evaluated into content. The specification of presentation defines how the content in the content specification is to be presented by a UIS (e.g., a WWW browser). It takes the form of a formal text in a language that the intended UIS can render (e.g., WML, XHTML, etc.). There is assumed to be a binding between the two specifications. Generally, such a binding may take the form of the renderable text being interspersed with variables referencing the content.

We define a hyperdocument to be a collection of hyperpages whose formal properties enable certain navigational operations to be performed over it. For example, with reference to an unordered collection and one of its' members, one can only request another member. If, however, the collection is known to be totally ordered, then requests for the next and previous member are meaningful. Clearly, the issue of designer-imposed structure on collections of hyperpages is a very important one, but it is also orthogonal to the architecture and therefore one that is not further addressed in this paper.

2.2 Meta Data

The purpose of meta-data is to describe the content and/or behaviour of a hyperpage or its' component parts. One can induce from a definition of a hyperpage (and recursively, its' component parts) a set of meta-data possibilities. Such meta-data may take the form of notes, generated by authors, users or the system. Generally, such notes are user-generic attributes of interest (e.g., the level of difficulty of a hyperpage is high) or instructions on how to tailor a renderable text. Meta-data can also be used to represent more abstract concept of links, tours and other structures.

The level of granularity that a hyperpage is subdivided by will determine the set of meta-data possibilities. For example, a flat hyperpage may only be annotated at the hyperpage level, while a hyperpage comprising of a sequence of information units enables annotation at the level of an unit or one of its component parts (e.g., a link).

Within this architecture, it is assumed that both hyperpages and their associated meta-data are stored in a *hyperlibrary*. It is assumed that the implementation of such a library would provide the functionality of a modern database system (e.g., associative querying, scalable retrieval and versioning).

2.3 The User Model Component

A *user model* may be viewed as a store of an individual users' information goals and history. Such a model is the basis upon which users may provide input to the adaptive process. User models can take many forms [4, 14, 15] and many different techniques have been used to acquire information about users [17, 16].

A user model may provide the means for a user to feed preferences into the adaptive process when it is system-initiated. The model reflects a users' goals and history. User modelling generally involves eliciting details from the user overtly or covertly. These details are then used by the adaptive mechanism to perform system-initiated tailoring. There are various types of user modelling, such as preference analysis [13], stereotyping [5] and activity analysis [2].

2.4 The Engine Component

The architecture within the core, depicted in Figure 2, shows two distinct groups of components. The *personalisation mechanism*, is responsible for providing the functionality required for user-initiated tailoring (personalisation) of hypermedia structures. The *inference mechanism*, is responsible for providing the additional functionality required for system-initiated tailoring (adaptivity).

The dynamics of the personalisation mechanism may be understood as follows. The personalisation process starts with a personalisation request, conveyed by a user, to the personalisation mechanism via a UIS. A personalisation request specifies a scope across hyperpages, or meta-data and the actions that the user wishes to effect. On receiving this request the *composer & tailoring engine* parses it to determine which hyperpages should be personalised and how, thereby generating a *personalisation program*. This program is a formal set of instructions, in a language known to the tailoring engine, that, when interpreted, retrieves the identified hyperpages, carries out the actions specified in the personalisation request and generates their user-specific versions, which are then stored.

Adaptivity is viewed as the process of allowing the system to take the initiative in personalisation actions, in the light of the systems' inference of a users' information goals and history (i.e., user-model). For adaptivity, a user-model and decision-making algorithm (containing a set of rules) are required. The role of these components is to initiate personalisation actions, possibly after consulting the user. When the hypermedia system identifies an opportunity to adapt (e.g., a request for a hyperpage) it does so by generating an adaptation request, which is a system-initiated personalisation request that carries out an action which the user may have issued, were he or she motivated to do so.

Using the architecture described, system-initiated personalisation is, in principle, as expressive as user-initiated personalisation and requires no functionality other than that provided by the inference mechanism. We argue that, using this architectural approach the components of this inference

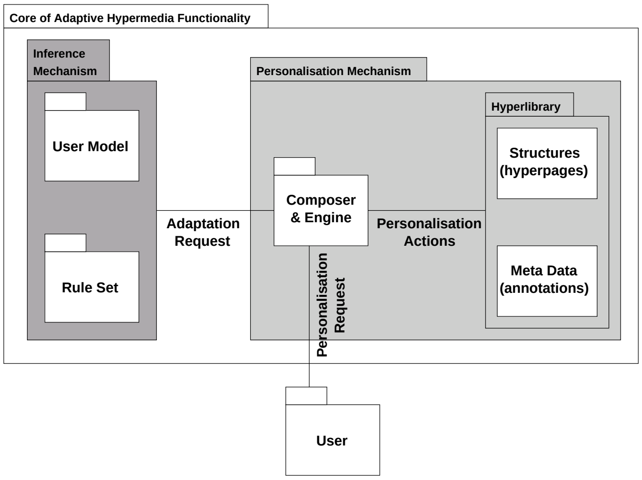

Figure 2: An Architecture for Adaptive Hypermedia

mechanism may be treated as “plug ’n’ play” black boxes that can be non-disruptively added to an AH system once an appropriate API has been developed.

The additional dynamics provided by the inference mechanism are as follows. The adaptive process starts when an event (e.g., a request for a hyperpage) is detected by some monitoring system. As a result of the detection of an event the inference mechanism consults the user-model (i.e., a store of an individual users’ information goals and history) to determine its current state. Using this knowledge the decision-making algorithm component applies a set of rules to each goal in the user-model. If a rule is satisfied then a set of rules and the user-model are used by the decision-making algorithm to construct and then suggest to the user a well formed personalisation request. This request is then submitted to the personalisation mechanism.

It is assumed that the core of AH functionality can also handle requests for hyperpages. Upon receiving a request, the composer component parses the hyperpage specification into a series of instructions that, when executed, convert the hyperpage specification into a renderable text (hyperpage), that is then returned to the UIS that initiated the request.

Note that since we are drawing an architecture within which the operation of AH systems in general may be understood, we do not specify the formal structure or semantics of personalisation requests and hyperpage specifications. To do so would limit the architecture to specific forms of hypermedia structures and specific languages for their manipulation. Furthermore, no constraints are imposed, regarding the choice of rules, user-model and decision-making algorithms, other than that they must exhibit the dynamic behaviour described in this section.

Using the architecture described in this section we now show how the AHAM, FOHM and GAHM approaches to modelling hypermedia may be represented within this architecture.

3. AHAM

One of the first formal models for adaptive hypermedia, the Adaptive Hypermedia Application Model (AHAM) [27, 10] builds on the earlier models formulated for more traditional hypertexts. AHAM is designed around an extended version of the DEXTER model [11]. DEXTER separates the components of a hypertext system into three major layers; the within component layer which stores the contents of the domain, the storage layer which contains the structure (nodes and links) between objects in the component layer, and the runtime layer which presents the hypertext information to the user. The DEXTER model also includes an anchoring layer to allow addressing of individual chunks of data within the component layer, and a presentation specification layer which provides the runtime layer with information on how to present specific hypertext components. The AHAM extension to DEXTER allows it to support adaptive hypermedia applications by separating the storage layer into a domain model, a user model and an adaptation model. This allows AHAM to provide a formal model for expressing adaptive hypermedia applications at the abstract level.

3.1 Hypermedia Structures

Much like our notion of hypermedia structures, AHAM’s domain model uses concept components to represent the abstract representation of an information item in an adaptive hypermedia domain. The structure of a concept is broken down into a set of attribute-value pairs, a sequence of anchors and a presentation specification.

To form a hypermedia, concepts are arrange in a directed acyclic graph. Atomic concept components represent a single fragment of information and their anchors reference the physical information, while composite components use a ’children’ attribute to specify a sequence of smaller composite components or atomic concepts.

As in the Dexter model, the raw data is stored in the within-component layer and all concept anchors reference the data in this layer. Presentation specifications determine how the particular data is to be displayed/rendered, although their application-specific nature means they are not modelled by Dexter or AHAM.

3.2 Meta Data

AHAM’s Meta-data, in the form of attribute-value pairs can be associated with both atomic concepts and higher-level composite components. At the hypermedia structure level, these storage units provide a means for describing the relationship types between concepts. AHAM also specifies a user model, overlaid on top of the domain model, to determine factors and actions that affect the user. The user model is also a set of attribute-value pairs that can be used to represent user-centric meta-data such as the required knowledge level of a user, or the status of a concept (read, ready to be leaned,). AHAM does not specify the complete set of possible meta-data values but allows authors to produce their own. AHAM also does not allow the ability to dynamically add new attributes-values, such as annotations, at runtime by the users of the system.

3.3 Engine Component

To combine the hypermedia structure and meta-data (or in AHAM terminology, domain and user models) AHAM

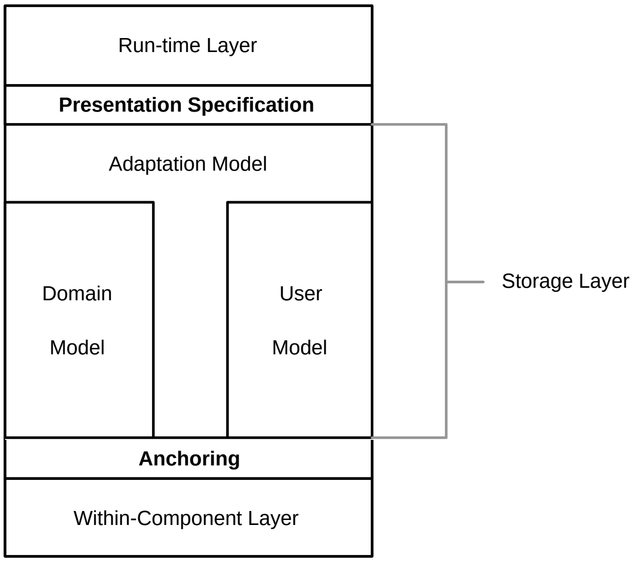

Figure 3: The Architecture of AHAM

uses an adaptation model which contains a set of adaptation rules, and an interpreter (or engine) to process these rules. Adaptation rules, written by a system designer, are stated in the form of event-condition-action clauses which provide the required mechanism to initialise the user model, update the user model and generate instances of adapted information.

4. FOHM

Work at the University of Southampton, has concentrated on analysing the fundamental components and structures of hypermedia systems. This work is part of the larger open hypermedia communityc̃ite1wsoh:94, 3wsoh:97,4wsoh:98,5wsoh:99 which have developed formal models for representing the structure and associations that exist within the underlying data components of hypermedia systems. To this end, a new open hypermedia model was developed; the Fundamental Open Hypermedia Model (FOHM) [21]. FOHM is largely based on the prior work with the Open Hypermedia Protocol (OHP) [8] which was designed to provide a reference model and architecture for Open Hypermedia systems. OHP placed an emphasis on the different structures belonging to hypermedia domains and raised the issue of how context might affect such structures. FOHM extends these ideas by developing a generalised model to represent the structure of these domains, and then provides the facility to attach context and behaviour objects to the model at various locations.

While FOHM provides a flexible theoretical model of structure which can be used to build hypertexts, an engine, Auld Linky [20], is required to instantiate and process the model. Auld Linky stores a database of FOHM objects (in XML format) and responds to queries from client applications for FOHM structures. Although FOHM was originally designed from an Open Hypermedia perspective, its application within the adaptive hypermedia field has been presented in [3].

4.1 Hypermedia Structures

The primary structures in FOHM are the data item and the association. Following earlier hypertext models, data items are attached to associations using a process of reference.

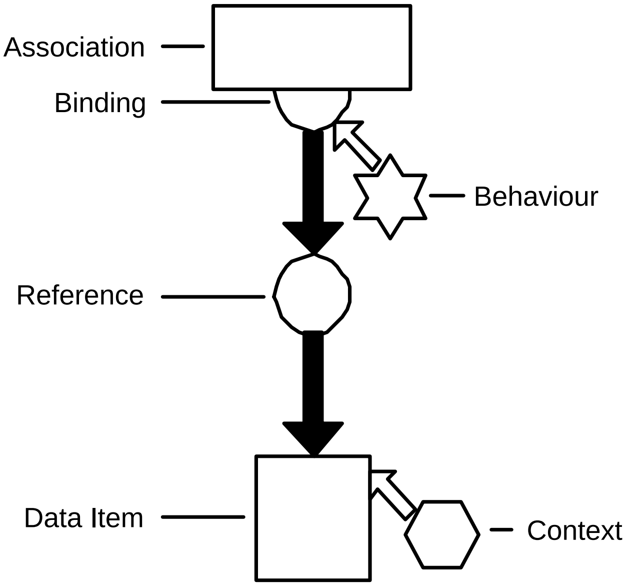

Figure 4: The Structure of a FOHM Object

Data objects are components that encapsulate a piece of information. Associations are links that relate together data objects and other associations. By combining these structures together, FOHM can support highly complex hypertext domains. During FOHM's development, several common arrangements of FOHM objects have been identified. *Tours* provide a sequential path across a set of objects, *Level's of Detail* are tours linking together increasingly detailed information and *Concepts* are associations that relate the same conceptual information using different presentation styles (i.e. handling different media representation of the same data). These structures can then be combined and arranged to suit a variety of hypertext domains.

4.2 Meta Data

To enhance the power of FOHM, two additional objects, context and behaviour can be used as meta-data/annotation components. They are implemented using attribute-value pairs (in a similar manner to the attribute-value meta-data in AHAM).

Context objects provide a means of limiting, or scoping, the current 'view' of the FOHM model. With this technique, a context object is attached to a FOHM query and it acts as a modifier, restricting the set of available FOHM objects that can be provided to the subset which have valid matching contexts. In an adaptive hypermedia domain, context objects can be used to represent restrictions on user views of a domain. This is analogous to the technique of user modelling, since a context object can be used to represent the current level of user understanding in a given subject.

Behaviour objects provide an event driven mechanism for specifying a set of actions. For example, a behaviour object can be attached to the 'on traversal' event of an association (such as a standard hyperlink) to specify the changes to the state of the system after the user has activated the link. For an adaptive hypermedia system, behaviour objects can be used to as a means of updating user models with new information based on the actions taken by the user.

4.3 Engine Component

The engine component of FOHM is realised by Auld Linky. Auld Linky manages a hypertext domain model marked up in XML as FOHM objects. When a client sends a personalisation request to Auld Linky (in the form of a FOHM association query) they can chose to optionally provide a Context Object. Auld Linky analyses the domain model to find parts that match the query pattern. It then collapses parts of the structure that fail to match the supplied Context. In this way, clients receive a contextual (personalised) view of the FOHM domain.

5. GAHM

The Goldsmiths Adaptive Hypermedia Model(GAHM) [23, 24, 25], developed at Goldsmiths College, University of London, takes a formal approach to the modelling of personalisable, adaptive hyperlink-based systems. The main contributions of which are an abstract model of core hypermedia-based functionality and, secondly, the definition of an abstract model of personalisability extending the core. Furthermore, an architecture is drawn for adaptivity using active rules as the decision-making component.

The model is comprised of three groups of functions: H-Region functions that model non-personalisable hypermedia-based interaction; P-Region functions that model user-initiated tailoring of hypermedia content; A-Region functions that model system-initiated tailoring of hypermedia content.

Within the H-Region, hyperpages are modelled as formal specifications. The semantics of hyperpage specifications are given with reference to an abstract machine whose operation and instruction set is formalised. The abstract machine is used to illustrate the execution of hyperpage specifications, thereby yielding renderable hyperpages.

Induced from the formal definition of hyperpage specifications is a set of annotation possibilities and personalisation & adaptation (P&A) actions. These enable all design decisions realised as hyperpage specifications to be revised.

Within the P-Region, personalisation is modelled as the user-initiated process of annotating and rewriting a hyperpage specification into a version that is associated with the user who took that action. It follows that the hyperpages users see may reflect their user-model if they wish.

When personalisation functionality is non-disruptively added to the H-Region, a designer can annotate a hyperpage in preparation for differences in users' goals and histories. A user can request to personalise both hyperpages and their annotations. Personalisation requests allow users to specify which hyperpages are to be personalised and how they should be transformed.

Hyperpage annotations and personalisation requests are modelled as formal specifications and a formal language has been defined for this purpose. Set-theoretic and relational algebraic expressions are used to represent the semantics of personalisation requests.

Within the A-Region, adaptivity is modelled as the process of allowing the system to take the initiative in personalisation actions. When adaptivity functionality is layered over H- and P-Regions, both users and designers can define strategies as to when the system should take the initiative and actively tailor interaction.

```
page{
 chunk{
  entry{
    [<a name="Intel 80386"></a>]  }
  content{
    A := 'Intel 80386',
    B := '(Commonly abbreviated to "386", trademark
 "Intel386") The successor to the',
    C := 'Intel 80286',
    D := 'microprocessor',
    E := '.  It was the first Intel processor with
 32-bit data and address',
    F := 'bus',
    G := 'es.  It can address four ',
    H := 'gigabytes',
    I := ' (2^32 bytes) of memory; however, 16
 megabytes is a typical maximum in'
    J := 'IBM PC'
    K := 's.'
  rendering{
    [<h2>                              A </h2><p>] B
    [<b><a href="80286.html">          C </a></b>]
    [<b><a href="microprocessor.html">D </a></b>] E
    [<b><a href="bus.html">            F </a></b>] G
    [   <a href="gigabyte.html">       H </a>    ] I
    [   <a href="ibmpc.html">          J </a>    ] K }}
 chunk{
  entry{
    [<a name="Intel 80386"></a>]  }
  content{
    A := 'select definition
          from   glossary
          where  term = 'i386'  }
  rendering{
    A  } } }
```

Figure 5: An Example Hyperpage Specification

In summary, the model is an abstract model, as many steps removed from concrete implementations as necessary to allow a systematic, exhaustive investigation of P&A issues in hypermedia systems. The model is an open model, insofar as hypermedia systems are viewed as clients of a variety of servers, in particular data and user-interface servers. Personalisation involves a transfer of ownership of the process of interaction with a hyperdocument from designers to users. To ensure that the set of personalisation actions is consistent, its elements are induced from the formal definition of the hyperdocuments they act upon.

5.1 Hypermedia Structures

Within the GAHM, hyperpages are defined to be a sequence of chunks, each of which is comprised of a *content specficiation* (C-Spec) and a *rendering specfication* (R-Spec). A C-Spec may take the form of data values or requests to DBSs for data values and may be associated with a set of *template* *variables*. Conceptually, a template variable is a placeholder for the content denoted by the C-Spec.

An R-Spec defines how content is to be rendered. It takes the form of a formal language that the intended UIS can render (e.g, HTML). This renderable text is interspersed with template variables, acting as placeholders for the content defined by the C-Spec. A chunk may be associated with a set of *entry points*, enabling it to be referenced, and a set of *exit points*, allowing for the establishment of a navigable link to another hyperpage or chunk.

A hyperdocument is defined to be a collection of hyperpages whose topology enables navigation between them.

5.2 Meta Data

The kinds of personalisation actions modelled by GAHM are based on the annotating and rewriting of hyperpage specifications. Annotation pairs a hyperpage specification with notes of interest to the user. These notes allow the assignment of values to attributes of the hyperpage and also allow the specification of rewriting actions over renderable texts (composed hyperpages). The existence of annotations facilitates the personalisation of a specified hyperpage and the recording of information about a hyperpage.

5.3 Engine Component

In GAHM, personalisation is viewed as the process of handing over to the user the ability to annotate, and/or rewrite, hyperpage specifications. Users can personalise both hyperpage specifications and annotations. A *personalisation request* is an editing command over hyperpage specifications that causes a modified version of a hyperpage to be versioned by the user who issued the request. A personalisation request specifies which hyperpages to personalise and what form to change them into. It is, therefore, a request to override the original decisions of the designers of a hyperpage (and of course, past expressions of preference by the user). A few examples of personalisation requests that a user might issue to the hyperdocument, of which Figure 5 is a page, are given in Figure 6.

Example 1 in Figure 6 is a personalisation request to tailor content. Its' effect is to insert the string "These pages must be revised" into the rendering expression of the first chunk of all hyperpages. **Example 2** is a request to add content. This request is applied to all hyperpages that contain the string "Electronic Mail" in the 5th chunk. **Example 3** is a request to rewrite all occurrances of the string "www.gold.ac.uk" to "www.goldsmiths.ac.uk". Its' scope is all hyperpages.

Within the GAHM, adaptivity is modelled as system-initiated personalisation. The GAHM approach to adaptivity centres on adaptive function. This function implements an inference engine over a decision theory, specified as a set of active rules, that describes which actions are more likely to yield the most benefits given a user-model. The function is responsible for suggesting personalisation actions, as described above.

6. DISCUSSION

```
select-page-if              % Example 1
  true
hp-then-do {
  insert [1, chunk (R-spec)]
     "These pages belong to student X"  }
                            % ---------
select-page-if              % Example 2
  [5, chunk] contains "Electronic Mail"
hp-then-do {
   insert [1, chunk]
     chunk {
       content  { X := 'Many of the latest
                       electronic mail systems
                       now provide support for
                       sound and video files.'  }
       rendering { [<I>] X [</I>]} } }
                            % ---------
select-page-if              % Example 3
  true
ann-then-do {
  insert page :
       "http://www.gold.ac.uk.com"
    -> "http://www.goldsmiths.ac.uk";  }
                            % ---------}
```

Figure 6: Personalising Fig. 5

The development of many AH systems has primarily been driven by technological innovation and constrained only by technical feasibility. The result of this hands-on approach has meant that development has rarely been accompanied by an abstract understanding of the components required for an AH system.

Due to increased interest in the use of personalisation and adaptation technologies, in the development of hypermedia applications, we believe that the hypermedia community is likely to benefit from a clearer consensus of how strong foundations for such applications may be laid.

The architecture proposed is purposefully cast at a level of abstraction above that of concrete systems, which has allowed for the representation of alternative models of AH. Using the architecture proposed, we have been able to clearly model the functionality of the components of AH systems in terms of its hypermedia structures, meta-data and engine component. By doing this, we have shown that additional AH components, such as user-models and decision-making algorithms, can be treated as interchangeable components of an AH system.

Our architecture also demonstrates how the expressiveness of P&A actions is ultimately determined by the structural approach chosen to define the hyperpages. The level of granularity that a hyperpage is subdivided by will determine the set of meta-data possibilities. In turn these semantic references to the content are the constraining factor when devising an engine for the manipulation of hyperpages.

Through the comparison of three qualifying models of AH, using the proposed architecture, we have shown that although taking different perspectives, all three address the same structural concerns:

1. the formulation of tailorable hypermedia structures,
2. the use of meta-data to provide semantics, or context, for these structures,
3. the development of mechanisms of hypermedia structures and their associated meta-data.

However, there are several noticeable differences between the three models that would have an impact on developers when choosing an architecture to model an AH system. Firstly, FOHM, while flexible and expressive, models only the personalisation mechanism aspect of an AH system. FOHM provides all the required hyperlibrary functionality while Auld Linky acts as the composer and engine. However, to be used within an adaptive environment, it needs to be coupled with these "plug 'n' play" components previously discussed. Unlike FOHM, GAHM and AHAM both support the personalisation mechanism and the inference mechanisms needed to model the core AH components.

Another difference that will be important to system designers, is that AHAM has been designed to operate on predefined data. In other words, AHAM does not handle dynamically generated data or meta-data at run time. This restriction was imposed to secure full knowledge of the adaptation rules at design time and therefore guarantee that all rules terminate, or at least identify those that do not. However it then limits the ability of any AHAM-based systems to create annotations on hypertext objects (such as pages or chunks) by users and then offer personalisation actions based on this meta-data.

In contrast to FOHM and AHAM, from a developers perspective, the GAHM is a functional model. Its' expressiveness is in its ability to clearly represent the functions required to compose hyperpages together with those required to tailor them.

GAHM illustrates the process of adaptation through the specification of an adaptive engine utilising a rule base and a formally defined user model. The adaptive function is formalised as an interpreter of active rules. Its' operation is based upon the dynamic updating of a users' history and information-seeking goals. However the GHAM's language for personalisation, although rich, is tightly coupled to the structure of its' hyperpages. As such, it may not be applicable in differing settings.

A final point to note is that, due to the functional nature of the GAHM, a degree of respecification may be required by a developer wishing to implement the GAHM using a modern object-oriented methodology.

It can be argued that, even though the architecture proposed in this paper could be presented in equivalent ways, the methodological procedure of isolating the various components and describing their high-level functionality and interactions is a contribution that may be used in other settings, under differing assumptions and using alternative conceptualisations of hypermedia systems.

7. CONCLUSIONS

In this paper we present a proposal for a component-based architecture that can be used to illustrate the components required for adaptivity. It is shown that, through an understanding of the structures that underlie these components, it is possible to devise an architecture in which personalisation and adaptation are clearly defined.

We have also shown that structure-oriented adaptive hypermedia models, AHAM, FOHM and GAHM, can be mapped onto this architecture. Although each model uses different representations we believe that there are important underlying commonalities.

This architecture has, therefore, provided a mechanism by which we can express these commonalities for P&A, i.e., hypermedia structures, meta-data and the engine component. Through an understanding of these common components and their relationships, we believe that it is possible for the AH community to devise a standardised "plug 'n' play" architecture. Such an architecture will enable the rapid development of future AH systems, by avoiding the reengineering of common components.

In addition, the identification of the commonalities between models of AH provides the first step towards the development of a set of services for communication between various AH components. Such a set of services could form the basis for a consensus as regards standards for interchangeability and interoperability between components of AH systems.

8. REFERENCES

[1] C. Bailey. *An Agent-Based Framework to Support Adaptive Hypermedia*. PhD thesis, University of Southampton, 2002.

[2] C. Bailey, El-Beltagy, R. Samhaa, and W. Hall. Link augmentation: A context-based approach to support adaptive hypermedia. In *Hypermedia: Openness, Structural Awareness, and Adaptivity*, 2001.

[3] C. Bailey, W. Hall, D. E. Millard, and M. J. Weal. Towards open adaptive hypermedia. In *Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web Based Systems*, pages 36–46, 2002.

[4] G. Brajinik, G. Guida, and G. Tasso. User modelling in intelligent information retrieval. *Information Processing and Management*, 23(4):305–320, 1987.

[5] P. Brusilovsky. Methods and Techniques of Adaptive Hypermedia. *User Modeling and User-Adapted Interaction*, 6(2-3):87–129, 1996.

[6] P. Brusilovsky. Adaptive Hypermedia. *User Modeling and User-Adapted Interaction*, 11(1-2):87–110, 2001.

[7] P. Brusilovsky, E. Schwarz, and G. Weber. ELM-ART: An Intelligent Tutoring System on World Wide Web. *Lecture Notes in Computer Science*, 1086:261–269, 1996.

[8] H. Davis and A. Lewis. A draft proposal for a standard open hypermedia protocol. In U. Wiil and S. Demeyer, editors, *Proceedings of the 2nd Workshop*

on Open Hypermedia Systems Hypertext '96, Washington D.C., 16-17 April 1996.

[9] P. De Bra, A. Aerts, D. Smits, and N. Stash. AHA! meets AHAM. In P. D. Bra, P. Brusilovsky, and R. Conejo, editors, *Adaptive Hypermedia and Web-based systems: Second International Conference, AH 2002*, LNCS, pages 213–222, Malaga, Spain, May 29 - 31 2002. Springer.

[10] P. de Bra, G.-J. Houben, and H. Wu. AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. In *Proceedings of the 10th ACM conference on Hypertext and Hypermedia*, Darmstadt, Germany, February 21-25 1999. ACM.

[11] F. Halasz and M. Schwartz. The Dexter hypertext reference model. In *Proceedings of the Hypertext Standardization Workshop*, volume 500–178 of *NIST Special Publications*, pages 95–133, Gaithersburg, MD 20899, January 1990. National Institute of Standards and Technology.

[12] H. Hohl, H.-D. Boecker, and R. Gunzenhaeuser. Hypadapter: An Adaptive Hypertext System for Exploratory Learning and Programming. *User Modelling and User-Adapted Interaction*, 6(2-3):131–156, July 1996.

[13] K. Höök, J. Karlgren, A. Waern, N. Dahlback, C. Jansson, K. Karlgren, and B. Lemaire. A glass box approach to adaptive hypermedia. *User Modeling and User Adapted Interaction: Special Issue Adaptive Hypertext and Hypermedia*, 6(2-3):225–261, July 1996.

[14] J. Karlgren, K. Höök, A. Lantz, J. Palme, and D. Pargman. The glass box user model for filtering. In *Proceedings of the 4th International Conference on User Modeling*, pages 221–222, Bedford, MA, USA, Aug. 1994.

[15] J. Kay and B. Kummerfeld, editors. *Proceedings of the Workshop on Adaptive Hypertext and Hypermedia User Modelling '94*, Cape Cod, August 1994.

[16] A. Kobsa. User modelling: Recent work, prospects and hazards. In *Adaptive user interfaces: Principles and pratice*, pages 111–128. Amsterdam: North-Holland Elsevier, 1993.

[17] A. Kok. A review and synthesis of user modelling in intelligent systems. *The knowledge Engineering Review*, 6(1):21–47, 1991.

[18] D. Lowe and W. Hall. *Hypermedia and the Web An Engineering Approach*. Wiley, 1999. ISBN 0-471-98312-8.

[19] N. Mathe and J. Chen. User-centered Indexing for Adaptive Information Access. *User Modeling and User-Adapted Interaction*, 6(2-3):225–261, July 1996.

[20] D. T. Michaelides, D. E. Millard, M. J. Weal, and D. C. D. Roure. Auld linky: A contextual open hypermedia link server. In *Proceedings of the Seventh Workshop on Open Hypermedia Systems*, pages 52–64, 2001.

[21] D. E. Millard, L. Moreau, H. C. Davis, and S. Reich. Fohm: A fundamental open hypertextmodel for investigating interoperability between hypertext domains. In *Proceedings of the ACM Conference on Hypertext, May 30 - June 3*, pages 93–102, San Antonio, TX, 2000.

[22] M. H. Ng, W. Hall, P. Maier, and R. Armstrong. The application and evaluation of adaptive hypermedia techniques in web-based medical education. *Association for Learning Technology Journal*, 2002.

[23] J. Ohene-Djan. *A Formal Approach to Personalisable, Adaptive Hyperlink-Based Interaction*. PhD thesis, Department of Computing, Goldsmiths College, University of London, University of London, 2000. Available from http://homepages.gold.ac.uk/djan/jodthesis.pdf.

[24] J. Ohene-Djan. Ownership transfer via personalisation as a value-adding strategy for web-based education. In P. Brusilovsky, N. Henze, and E. Milln, editors, *Proceedings of the Workshop on Adaptive Systems for Web-based Education, held in conjunction with 2nd International Conference on Adaptive Hypermedia and Adaptive Web Based Systems*, Mlaga, Spain, May 28 2002.

[25] J. Ohene-Djan and A. Fernandes. Modelling personalizable hyperlink-based interaction. In P. Brusilovsky and P. De Bra, editors, *Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia (Ninth ACM Conference on Hypertext and Hypermedia HYPERTEXT'98)*, pages 87–91, Pittsburgh, USA, June 20-24 1998. ISSN 0926-4515.

[26] J. Vassileva. A task-centered approach for user modeling in a hypermedia office documentation system. *User Modeling and User-Adapted Interaction*, 6(2-3):185–224, 1996.

[27] H. Wu. *A Reference Architecture for Adaptive Hypermedia Applications*. PhD thesis, Techniche Universiteit Eindhoven, November 2002.