Family Resemblance for Hypermedia Authoring

N. Bryan Kinns

April 1999
Family Resemblance for Hypermedia Authoring

N. Bryan-Kinns
Department of Computer Science
Queen Mary & Westfield College
University of London
Mile End
London, E1 4NS
nickbk@dcsc.qmw.ac.uk
April 1999

Abstract

This paper introduces the concept of family resemblance to hypermedia authoring. It develops schema based techniques which define the structural and semantic regularities of classes of hypermedia applications to support the explicit definition of similarities between schemas. Moreover, it uses these similarities to support propagation of changes between schemas and applications created from them. This development reduces the authoring effort involved in hypermedia application construction, and increases the appeal of schema based approaches.

Keywords

Hypermedia, Hypermedia authoring, Family resemblance, Reuse.

1 Introduction

Hypermedia applications are often inconsistently structured and have poorly defined semantics [6]. This may be intentional (e.g. artistic endeavours) or may be the result of the typically ad-hoc nature of hypermedia application development. The results of such inconsistencies and ambiguities are often user disorientation [6], and increased authoring effort [9].

Several approaches have addressed this problem by supporting the description of the structural and semantic regularities of hypermedia applications e.g. HDM [8], HyperStorM [1], and VCMF [4][5]. Once these regularities have been defined for a class of applications (typically by a schema) individual applications are developed. A schema usually contains node types related to each other by link types. These define the semantic and structural regularities of applications respectively. Node types define the kinds of nodes (units of information), their attributes and possible values in applications. Link types define the kinds of links that can exist in applications, and constrain which nodes can be linked.

Unlike unstructured approaches, using a schema based approach means that applications will have consistent structures and semantics [12] which should reduce user disorientation [6], especially when the structure matches the user’s expectations. Moreover,
the approach can be used to reduce authoring effort by automatically generating values for nodes and creating certain kinds of links [8]. Such approaches have been further developed by VCMF [4][5] which uses the notion of schemas to support reuse of hypermedia information (video in the case of VCMF).

Although the use of schemas can reduce the effort of creating hypermedia applications, the effort involved in creating schemas themselves, and in particular creating related variations of schemas can be off-putting. This paper argues the case for the introduction of family resemblance which supports many similar variants of the same schema thus reducing authoring effort in such cases.

The rest of this paper is organised as follows. First the problem addressed is detailed. This is followed by a solution to the problem, and an example of the use of the solution. Approaches to supporting the solution are then discussed. The paper is concluded with suggestions about its integration with other approaches and a summary.

2 The Problem: Poor Support for Authoring Related Schemas

Recent work using VCMF [4][5] shows that using a schema based approach to authoring hypermedia applications (in this case applications which contain nodes of video and associated annotations, as well as links; hypervideo applications) supports the definition of the structural and semantic regularities of hypermedia applications. Moreover it shows its support for reuse of nodes and annotations between applications by reference which was previously difficult.

However, the work also identified an issue with schema based authoring: often many similar versions of the same schema are required to develop related applications. This issue has not been addressed in previous work such as HDM [8] and HyperStorM [1] as they have concentrated on the creation of one hypermedia application from one schema. In recent VCMF work five closely related schemas were used to create three hypervideo applications. These schemas were frequently amended, and these amendments were reflected in related schemas. Currently hypermedia systems cumbersomely support the development of related schemas by making copies of schemas with the appropriate amendments. This means that changes in one schema are not reflected in schemas derived from it which increases authoring effort and inhibits reuse of schemas in such situations.

Increased schema authoring effort reduces the attractiveness of schema based approaches. This is because it is not easy to update hypermedia applications created from schemas, nor related schemas. A result of these authoring overheads is that schemas tend to be static; they are typically not updated once defined. This in turn tends to mean that schema based approaches are typically used in situations where schemas are well defined beforehand e.g. technical documentation where the structures and semantics are well defined [8]. Many situations require more flexible schema authoring e.g. systems supporting sociological data analysis need to allow authors to refine schemas (in particular coding schemes) as the data is analysed [16].
3 Solving the Problem: The Introduction of Family Resemblance

In order to address the issue of authoring related variants of schemas this paper introduces the notion of family resemblance (inspired by Rosch [14]) to schema authoring. Used here, family resemblance supports the explicit definition of the similarities and groupings of schemas, and can support the propagation of change through families of schemas. Moreover, it can be used to support the propagation of change from schemas to the applications created from them e.g. changes to attribute types in object types are reflected in the objects created from them.

The basic principle is that a family contains a set of named schemas which are related to each other in terms of their resemblances. Schemas are derived (refined) from a parent schema; these derivations constitute the family resemblances (there is one root schema which is not derived from any other schema; it typically defines the general structure and semantics of schemas in the family). These refinements involve some change in the content of the schema between the original and derived schemas. When a schema is changed, these changes are propagated to schemas derived from it (and applications created from them) as well as to applications created from it. This reduces authoring effort where several related schemas are used as they do not need to be manually copied and updated. Moreover it provides a representation of the relationships between schemas which can be useful for authors when conceptualising their work when many related schemas are involved. Some schemas (referred to as abstract schemas) may not be used to create applications.

Unlike object inheritance [13] which supports the creation of node types (classes in object-oriented terminology) from others (super-classes), family resemblance is concerned with the refinement of schemas from other schemas (which themselves contain many node and link types). Moreover, object inheritance involves creating new node types from others whereas family resemblance expresses the similarities between schemas; new schemas are not created. This paper proposes family resemblance as a more powerful form of derivation than object inheritance.

Furthermore, family resemblance subsumes schema evolution [3] which involves refining one schema and ensuring that nodes created from it are suitably updated. Using family resemblance not only are the nodes and links updated, but so are schemas related to the refined schema.

Family resemblance essentially takes the work of schema versions [10] and extends it. Schema versions usually only consider two versions of schemas: transient and working. Family resemblance involves the definition of many related versions of the same schema; family members.

The next section presents an example of the use of family resemblance to illustrate the concepts involved.
3.1 Family Resemblance in Use

Several hypervideo applications have been developed using VCMF to show its ability to model the structural and semantic regularities of current uses of digital video. In turn, several schemas have been created to construct these applications. These include schemas to support the analysis of plot in narrative films, the construction and use of video-maps (spatially organised hypermedia applications such as those discussed by Lippman [11] and Sawhney et al. [17]), and the analysis of style in narrative films (similar to the analyses which can be performed using Media Streams [7]). The issue of family resemblance is particularly striking in the use of schemas to support the analysis and construction of plot in films (plot structure schemas).

The first part of any academic analysis of the plot in a narrative film is the plot segmentation. This involves analysing the plot in terms of its sequential and compositional structures (non-narrative films, which are not story based, require different analysis techniques). In the terminology used in this paper, plot analysis involves creating a hypervideo application which represents the plot and its structure. The plot itself is represented by nodes of information detailing the content of plot elements. These are linked to each other to provide the sequential and compositional structures, and are associated with appropriate pieces of video.

Figure 1 shows the general structure of plot segmentation discussed by Bordwell et al. [2]; similar structures are evident in many video modelling systems such as Rowe et al.’s Video Database Browser [15]. Italic text in the figure describes the composition of the plot elements (plot elements shown by non-italic text). The structure is as follows (each plot element is given an identifier which reflects its sequential position in the plot e.g. Part 1, Scene 2a):

- **Parts** represent major divisions in the plot, and are composed of sequences of Sub-Parts and Scenes.
- **Sub-Parts** represent logical divisions of the plot within Parts, and are composed of sequences of Scenes.
- **Scenes** represent distinct phases of action occurring within a relatively unified space and time (with respect to the plot), and are composed of sequences of Sequences and Shots.
- **Sequences** represent logical divisions within Scenes, and are composed of sequences of Shots.
- **Shots** are uninterrupted sequences of video from one camera and form the lowest level of granularity of the plot.

This can be regarded as a schema describing the general structure and semantics of hypervideo applications which model plots of narrative films. It will often be necessary to slightly change the structure or content (i.e. node types) of this schema when performing an analysis of the plot of a film, or constructing a plot based film. In particular, the related schemas of Figure 2 illustrate the need to refine the general schema (the root in this family) in order to create a schema more suitable for the analysis of plot. In such analyses it can be
assumed that the nodes of the hypervideo application will be ordered according to the start frames of their associated video segments which are contiguous. This can be used to help automatically instantiate start frames of video related to nodes and to create appropriate temporal links between nodes, so reducing authoring effort. Moreover, identifiers such as Part 1, Scene 2a (discussed previously) can be automatically inferred from the nodes’ sequence which again reduces authoring effort. This derived schema is shown in the figure to the left and below of the general schema (the refinement is shown in italic text). Similarly, to the right and below is a derived schema whose purpose is to support the construction of presentations (in this case documentaries). Unlike the schema for analysis, this schema’s nodes are ordered by the user as they construct the presentation.

The analysis and construction schemas show two forms of resemblance: between and within (illustrated in the Figure 2). There are resemblances within the set of schemas derived from the general plot structure (both the analysis and construction schemas share similar structures). These can be regarded as sibling similarities. Similarly, there are resemblances between the analysis schema and the general plot structure as well as between the construction schema and the general schema. These can be regarded as similarities between parents and children.

The analysis schema is not necessarily suitable for all forms of plot analysis. For example, modelling the plot of Citizen Kane requires description whether Scenes in the plot are flashbacks or not. Flashbacks present events before the main period of the story. In Citizen Kane, the main period of the story is a reporter’s investigation into Kane’s life (Kane is the main character of interest in the film) and so flashbacks are used to present various character’s memories of Kane during his life. This can be supported by deriving a schema from the analysis schema which adds attributes to the Scene node detailing whether it is a flashback, and what date the flashback refers to. This is shown in the figure below the analysis schema. To the left of this schema is the actual hypervideo created from it; the plot segmentation of
Figure 2: Family resemblances of schemas

*Citizen Kane.* As this schema is derived from the analysis schema it benefits from being able to automatically infer identifiers for Parts etc. from nodes’ sequential positions.

In VCMF work the construction schema is used to (re)create a documentary about the film *Citizen Kane.* This is shown to the right of the construction schema in Figure 2. The work also derives a schema from this which is used to create documentaries about areas in video-maps (shown in the figure below the construction schema). Another schema is used to create video-maps themselves. Therefore nodes from that schema are included in this newly derived schema to facilitate reuse of nodes from video-maps by reference *i.e.* without copying the node into the new hypervideo; reuse by reference is a major benefit of using VCMF. To the right of this schema is the hypervideo generated from the schema which mixes narrative nodes with (reused) interactive video-map nodes.

The family illustrated in Figure 2 shows the utility of family resemblance for schema authoring in three main ways:

1. It is not necessary to manually copy schemas in order to refine them; instead the family resemblance defines the similarity between schemas. This saves authoring effort.

2. Any changes in schemas will be reflected in schemas derived from them *e.g.* changes in the general plot structure schema will be propagated to the schemas for construction and analysis. This makes schema authoring more flexible and iterative and means that no effort is involved in manually propagating changes between related versions of schemas.

3. It provides a representation of the similarities between schemas. This can be useful for authors when conceptualising their work.
4 Support for Family Resemblance

Algorithms to support family resemblance are not intractable; schema versions algorithms [10] can be adapted to support the appropriate derivation relationships and propagation between schemas. Furthermore, schema evolution algorithms [3] can be used to support propagation of changes from schema to applications created from them.

The most interesting issues lie in the HCI aspects of how users can construct, and be presented with, comprehensible family resemblance information. Current work has started to address this issue. Using the conventional notion of object-oriented class browsers such as those provided by Symantec’s Visual C@fe™ are not suitable as they typically show a small set of relationships between object classes and do not detail their similarities e.g. relationships for: inheritance, message passing, references to external classes. Tools which support family resemblance need to accept and display much richer information about relationships between schemas including:

- Modifications, additions, and deletions of node type attributes.
- Modifications of the orderings of nodes within other nodes i.e. the orderings of collections of nodes.
- Addition and removal of node types.
- Addition, removal, and modification of link types.

Figure 3 is a screen shot from a prototype implementation supporting family resemblance within the VCMF toolset (an extensible component based toolset supporting the use of schemas to create hypervideo applications [5]). The display shows family resemblance in a similar manner to Figure 2. However, the prototype raises many HCI issues which will be addressed by future work. These include:

- How to succinctly show the differences between schema. In the current implementation this is achieved by placing text on the line representing the resemblance. However, there may be many changes between each schema which causes a presentation problem (resolved in the prototype by allowing [more] to be selected to display the full list of changes).
- How users indicate the resemblance between schemas. Currently a schema is selected, a new version of it is created, modifications are made by the user, and the system then represents these modifications in the interface. Future work needs to investigate how users can succinctly and intuitively express the resemblance between schemas.
- In the current implementation schemas can be double-clicked to bring up a new window showing their node and link types. In this window modified node and link types (including additions) are highlighted. However, this does not display deletions, and moreover, it is difficult to visualise which derivations come from which schemas. The issue here is one of succinctly showing the changes between schemas in terms of nodes and links as well as their attributes.
5 Conclusions

In terms of hypermedia work, family resemblance could be usefully employed in approaches which support schema based authoring such as HDM [8], as well as the object-oriented based work of HyperStorM [1] which also supports the definition of structural and semantic regularities of classes of applications. Earlier work such as Catlin et al.’s Hypermedia Templates [6] would also benefit from the introduction of family resemblance. An issue identified in their work is that of hot templates; templates are similar to schemas, hot templates refer to the propagation of change from schemas to applications created from them. As discussed previously, family resemblance supports propagation of change from schemas to applications as well as between schemas.

This paper presented family resemblance as a solution to the problem of creating and using many related schemas when constructing hypermedia applications. Family resemblance supports the definition of groups of schemas, and the definitions of similarities between groups of schemas and individual schemas. Using this approach reduces authoring effort and so makes schema based authoring more attractive to developers. Moreover it provides a representation of the similarities which may help authors to conceptualise their schema based work. This paper showed the use of family resemblance to support the development of hypervideo applications from related schemas. Future work needs to investigate and address the HCI issues of manipulating family resemblances.

References


