

Conflict Management in Multi-model Adaptive Hypermedia

Luis Francisco-Revilla and Frank Shipman

Center for the Study of Digital Libraries and Department of Computer Science

Texas A&M University

College Station, TX 77843-3112, USA

+1 979 845-4924

{l0f0954, shipman}@csdl.tamu.edu

ABSTRACT

New adaptive hypermedia systems are employing multiple independent models in order to better guide their adaptation by considering relevant factors in addition to user characteristics. This approach promises enhanced functionality but it entails the possibility of conflicts as different models can suggest contradictory adaptations. Finding mechanisms capable of automatically managing these conflicts is a key issue in the development of the new generation of adaptive hypermedia systems. This work provides an approach that delivers a context-sensitive solution to this issue within the field of adaptive spatial hypermedia. The paper discusses the approach and its instantiation, including the underlying concepts, system architecture, adaptation process and key features.

CONTENTS

1. Introduction
2. Adaptive Spatial Hypermedia
3. Conflict Management
 - 3.1 Scenario of Use
 - 3.2 Conflict Prevention
 - 3.3 Conflict Detection
 - 3.3.1 Context of Conflicts
 - 3.4 Conflict Resolution
4. WARP
 - 4.1 Adaptation Process
 - 4.2 Conflicts and Constraints
 - 4.3 Key Implementation Features
 - 4.3 Approach Evaluation
5. Conclusions

1. INTRODUCTION

Authors composing new documents have to consider the needs of their target audience in order to ensure a proper communication of their ideas. However, in environments as prolific as the Web, audiences can vary greatly, making it difficult to determine the context of use for the documents. In these environments, documents are accessed for different purposes, in different contexts and by many people with very different backgrounds and interests. Presenting their ideas in ways that address the potential needs of everyone in audiences like these would require authors to create a multitude of documents. This quickly becomes prohibitive in terms of time and effort. Therefore, authors often opt to compose general-purpose documents that address the common needs of the largest segment of the envisaged audience.

Nevertheless, individual readers often have particular needs, interests and goals that are not completely met by general-purpose documents. Hence, readers often have to adapt to general-purpose documents, mentally filtering the irrelevant components and focusing on the relevant components. While most people are used to performing this activity in their everyday lives – for example crossing or circling ads when reading the classifieds section in a newspaper – it represents an overhead in their process of finding and using the information they need.

The field of Adaptive Hypermedia emerged as researchers began to focus on devising systems that would reverse this situation such that general-purpose documents would adapt to the individual reader rather than the contrary.

Adaptive hypermedia systems better support authors by facilitating the creation of documents useful to wider audiences. Furthermore, adaptive hypermedia systems help readers by reducing the effort required to identify the information relevant to them.

The typical approach used in adaptive hypermedia involves abstracting relevant user characteristics – such as interests, knowledge or preferences – and storing them in a user model. This user model is then employed to determine the appropriate modifications to the hypermedia documents – such as hiding or annotating paragraphs and hyperlinks (De Bra et al. 1999; Brusilovsky 1996).

Given that human actions are situated and depend heavily on their particular context (Suchman 1987), it was soon evident that, in addition to user characteristics, other factors – such as task and situation – needed to be considered as well. Consequently, researchers began expanding their user models with these additional factors. This resulted in more sophisticated models that improved the responsiveness of the systems. However, this also entangled the user characteristics with other unrelated factors increasing the complexity of the models.

A general interest in adaptive hypermedia research is to create systems capable of scaling and migrating through domains and applications (Encarnação 1997; Stephanidis et al. 1997). However, the use of a single complex user model often results in rigid adaptation mechanisms with limited ability to migrate and scale through applications, domains and time. In response to this, researchers began exploring the simultaneous use of user, task, and situation models for delivering medical information (Francisco-Revilla and Shipman 2000; Francisco-Revilla

1998) and, subsequently, the use of domain, task and user models for performance support (Brusilovsky and Cooper 2002).

The systems in this first generation of multi-model adaptive hypermedia were designed assuming a fixed set of models. This facilitated the coordination of the models in order to ensure a consistent overall functionality. However, this approach limits the systems' ability to scale and migrate since it is impossible to predict which and how many models will suffice for every domain and application.

An obvious enhancement to this first generation of multi-model adaptive systems is the use of multiple *independent* models that can be combined and reused as domains and applications demand. However, a challenge with this approach is that different models can suggest mutually exclusive adaptations. Hence, systems need to provide mechanisms that can manage these potential conflicts in ways that do not depend on the identity, number and nature of the models.

This paper presents an approach that delivers a context-sensitive solution to managing conflicts automatically within the field of adaptive spatial hypermedia. The presentation is organized in five sections. First, it provides an introduction to spatial hypermedia and in particular to adaptive spatial hypermedia. Next, it discusses the approach of conflict management (as opposed to conflict resolution). Subsequently, the paper discusses the instantiation of the approach, including the system architecture and the operation of the adaptation process. Finally, the paper wraps up with the conclusions.

2. ADAPTIVE SPATIAL HYPERMEDIA

Spatial hypermedia originated as the result of observations of how people use map-based hypermedia systems such as Aquanet (Marshall et al. 1991). These observations revealed that, rather than explicitly creating links (arrows) between objects, people often used only visual attributes and the relative spatial position between objects in order to imply relationships (Marshall et al. 1992; Marshall and Shipman 1993). Observations of how people organize their workspace in the physical world validated these observations (Malone 1983). This motivated the development of spatial hypermedia systems like VIKI (Marshall et al. 1994) and VKB (Shipman et al. 2001) that explored the use of space to represent explicit and implicit links with varying degrees of formality (Shipman and McCall 1994; Shipman et al. 1995).

Spatial Hypermedia systems provide a medium of expression that uses the relative position and visual characteristics of information objects to represent implicit and explicit relationships between them. It accomplishes this by providing a *space*, where *objects* are placed, making their *relationships* visually perceivable.

Reading spatial hypermedia documents often involves the moving, resizing, and/or changing of the information objects (Francisco-Revilla and Shipman 2004a). This allows readers to manually adapt the documents to their needs – an approach referred as adaptable hypermedia (schraefel 2000). In addition, this ability to interact with

the presentation of the information naturally supports the fluctuation between information sorting and information structuring exhibited by knowledge workers (Buchanan et al. 2004; Nakakoji et al. 2000).

The high expressiveness, interactive nature, and support of information workers, ambiguities and fluid formalization of the information, made spatial hypermedia particularly attractive for delivering information in domains where readers were interested in working with and reusing the information (as opposed to just causally reading it). This motivated the development of presentation-oriented spatial hypermedia.

The advent of presentation-oriented spatial hypermedia created new possibilities for the communication and use of information by both authors and readers. At the same time, it also changed some of the previous assumptions in spatial hypermedia revealing new challenges. For instance, while spatial hypermedia authors and readers were originally assumed to be the same group of people, this was not the case anymore in presentation-oriented spatial hypermedia. And, even though spatial hypermedia allows readers to manually modify the documents, this was often not enough. This led to the emergence of adaptive spatial hypermedia – just like adaptive hypermedia emerged from static hypermedia

Adaptive spatial hypermedia was informed by the approaches developed in adaptive hypermedia – such as the separation of adaptation methods and techniques (De Bra et al. 1999; Brusilovsky 1996). However, these approaches needed to be extended as spatial hypermedia poses additional challenges to adaptation. For example, the high expressiveness of spatial hypermedia complicates the adaptation process because it allows many more ways to represent the suggested adaptations. This, together with the need to consider additional aspects such as the space and implicit relationships between objects (intrinsic to spatial hypermedia and absent in navigational hypermedia), motivated the creation of a new approach that enabled the development of adaptive spatial hypermedia systems (Francisco-Revilla 2005; Francisco-Revilla and Shipman 2004b).

While several systems have been developed in order to explore the representation potential of spatial hypermedia (Grønbaeck et al. 2002; Hsieh and Shipman 2000; Marshall et al. 1994; Shipman et al. 2001; Web Squirrel), to the best of our knowledge, MASH (Francisco-Revilla 2005; Francisco-Revilla and Shipman 2004b) and WARP (Francisco-Revilla and Shipman 2004c) are the only work done in regard to augmenting spatial hypermedia with adaptive functionality, with the exception of the access constraints in HyperMap (Trigg and Weiser 1986). The general overview of the MASH approach (Francisco-Revilla and Shipman 2004b), WARP's implementation (Francisco-Revilla and Shipman 2004c) and the effects of multi-model adaptation in the reading process in spatial hypermedia (Francisco-Revilla and Shipman 2004a) have been presented elsewhere. This document focuses on presenting the particulars of how to adapt spatial hypermedia documents using multiple independent models and how to automatically manage the conflicts that can arise between the models' adaptation suggestions.

3. CONFLICT MANAGEMENT

Conflict management, as opposed to conflict resolution, is a comprehensive approach that encompasses the processes of: *conflict prevention*, *conflict detection* and *conflict resolution*. Conflict prevention aims to avoid conflicts altogether or to make conflict resolution unnecessary by using ambiguous presentations. However, given that it is not possible to sidestep every conflict, it is necessary to have a detection process that identifies conflicts in a timely manner and determines their scope. This seemingly straightforward process has its intricacies because conflicts can propagate across relationships. Finally, conflict resolution addresses conflicts by providing corrective actions or suggesting alternative representations for adaptations that do not generate conflicts.

These processes are better understood by considering them within a scenario of use. The following section provides such a scenario, which serves as the base for the subsequent explanation of how each of the processes work and how some particular features of spatial hypermedia features facilitate their operation. Sections 3.2, 3.3 and 3.4 describe what the challenges are, while the mechanics of how to address them are discussed in section 4.

3.1 Scenario of Use

Consider a system that takes various sources of news and then adapts and presents them as a spatial hypermedia document to individual stockbrokers that work in a brokerage firm. This presentation allows brokers to organize, analyze and understand the information by allowing them to move, change and annotate the objects.

The system employs a user model that identifies relevant news stories using information about the individual user's investment portfolio. The user model classifies articles or news stories as *very relevant* if they talk about a company in the broker's portfolio. Alternatively, when presented with a news story regarding an industry that encompasses at least one of the companies in the portfolio, the user model classifies it as *so-so relevant*, even if it does not talk about any of the companies in the portfolio. The model classifies all other stories as *not relevant*.

In addition to the user model, the system uses a competitor model that classifies the news stories similarly to the user model, although in this case the model bases its decisions on the market performance of a competing trader.

Finally, the firm's management included a risk model that classifies the news stories based on the confidentiality of the information and the risk that the use of this information might prompt accusations of insider trading. Depending on these factors, this model classifies news stories as *very risky*, *so-so risky*, and *not risky*.

Once the models classify the news stories, the system adapts their presentation by emphasizing the relevant and de-emphasizing the irrelevant ones. Similarly, news stories are annotated based on how risky they are. Finally, very risky news stories are protected from viewing.

Based on the functionality described above, consider the case of an article containing confidential information about a company that belongs to an industry in which the user does not operate but the competitor does. In this case the system needs to deal with suggestions for the same object to protect it from viewing, de-emphasize it and

emphasize it. The following sections reflect on situations like this as a way to explain the processes of conflict prevention, conflict detection and conflict resolution.

3.2 Conflict Prevention

In multi-model adaptive hypermedia (navigational or spatial), conflicts occur when different models suggest adaptations that cannot be represented simultaneously. As an example, consider the suggestions from the competitor model and the risk model mentioned above. Imagine that the system's only way to emphasize the news story is to highlight its text and the only way to protect the news story from viewing is to hide it. This obviously creates a conflict since it is impossible to highlight an object if it is hidden.

Conflicts can be avoided by designing models to suggest high-level adaptation methods capable of being mapped to various adaptation techniques. Consider the previous example where the competitor model suggests to "emphasize" (instead of highlighting the text of) very relevant news stories and the risk model suggests to "protect from viewing" (as opposed to hide) very risky items. This allows the system to select appropriate mappings for each adaptation method. For instance, "emphasizing" can be accomplished by changing the background color and "protecting the news story from viewing" can be achieved by occluding the text while keeping a placeholder for the object. Since both of these adaptation techniques can be simultaneously represented, the conflict is averted.

Supporting ambiguity in the presentation is another approach that, while it does not prevent conflicts from occurring, it sidesteps the need to resolve them. In the previous example, consider the conflict between the user model and the competitor model. One suggests de-emphasizing the news story while the other suggests emphasizing it. If these methods are implemented using complementary techniques (for instance increasing and decreasing the font size) then these adaptations will cancel each other out. Alternatively, in a system that supports ambiguity, each method can be expressed using a different technique. For instance, if de-emphasizing reduces the size of the object, then emphasizing can change the color of the text. This conveys the meaning that the news story is relevant from the point of view of one model and irrelevant from the other model's perspective. The decision of whether or not to support ambiguity is important and depends on the domain and application.

3.3 Conflict Detection

Detecting conflicts when different models propose contradictory adaptations for the same object is fairly simple. However, given that spatial hypermedia objects can be implicitly related, conflicts can be created indirectly as by-products of adaptations to other objects. For instance, in the adaptive news system, one object presented to the user can be a picture of a company's production floor while another object can be the text that comments the picture. In cases like this, hiding the picture can render the text meaningless.

Determining the scope of conflicts and how they propagate through the relationships between objects is challenging. Fortunately, spatial hypermedia systems have the ability to identify these implicit relationships using

their spatial parsers (Francisco-Revilla and Shipman 2005). This feature of spatial hypermedia systems provides a method for detecting indirect conflicts and helps to determine their scope.

3.3.1 Context of Conflicts

Spatial parsers are a key feature of spatial hypermedia systems used for the recognition of the implicit structures such as piles and lists that emerge as a result of the relative positioning and visual similitude of the objects (Francisco-Revilla and Shipman 2005). These structures are often used by authors to express meaningful relationships informally. Spatial parsers typically recognize structures by grouping sets of objects *into* composite objects, where composites “contain” atomic objects or other composites. This process iterates until the overall document structure is computed.

The hierarchical structure determined by the spatial parsers allows the system to determine the scope or context of conflicts – and how they propagate – as a function of containment. This in turn allows conflicts to be classified into the following categories:

- Object conflicts. These occur when an object receives some adaptation suggestions that oppose each other – e.g. when one model suggests hiding the object while another suggests showing it.
- Sibling conflicts. These occur when suggestions for an object affect or clash with the presentation of another object in the same context. For instance, suggestions to show and highlight a text that talks about a picture can conflict with the suggestion of hiding the picture.
- Parent-child conflicts. Parent-child conflicts occur when suggestions for the adaptation of the parent composite oppose suggestions for the components of the composite (children). For instance, suggestions of hiding a list of objects conflict with suggestions of highlighting individual objects in the list.
- Parent conflicts. Since composites are computed based on the spatial positioning and visual appearance of the objects, modifications to the visual appearance of an object can affect its parent composite potentially destroying it. For instance, hiding an object in a list of objects can visually break the list in two. While sometimes it might be best to maintain the two lists as separate objects, other times it might be better to “contract” the list, repositioning the other objects in such a way that it still is recognized as a list. Whether any corrective action is appropriate depends on each case. Parent-child conflicts differ from parent conflicts, in that the former are the result of contradicting adaptations at different levels of the document hierarchy, while the latter are by-products of adaptations for a single object.
- Parent-parent conflicts. Objects can often belong to multiple composites. For example, consider a vertical list of objects that intersects a horizontal list. The object in the intersection is contained in both lists. Hiding this object affects both lists. However, the parent lists are limited in the corrective actions that they might use. It is not possible to contract both lists as two objects would be “pulled” into the same position. Alternatively, instead of

hiding the object in the intersection, it might be better to replace it with a placeholder that, while occluding the information, still conveys the meaning that an object is there and that there are two intersecting lists.

3.4 Conflict Resolution

Since conflicts can propagate across object relationships – for instance, a sibling conflict can spread from sibling to sibling – it follows that conflicts must be resolved early before they can propagate far.

Depending on the situation and their type, conflicts need to be resolved differently. It is necessary to be flexible and support different resolution strategies. Consider the financial news system mentioned above. Assume that an object conflict is detected between the user model's suggestion of emphasizing a particular news story and the suggestion of the risk model to protect it from viewing. If the firm wants to minimize the chances of facing an insider trading lawsuit, then the best conflict resolution strategy might be to assign priorities to each model. This way the risk model suggestion can override the suggestions of the other models. However, an object conflict between the user model's suggestion of de-emphasizing an object and the suggestion of the competitor model to emphasize it might be best resolved differently. In this case, a better solution would be to always take the suggestion that classifies the news story with the higher relevance.

There has been previous research on the issue of competing suggestions offered by different mechanisms (Ferber 1999; Sandholm 1999; Sycara 1989). Different schemas have been suggested, such as voting, priority-based, and market-based strategies (Conry et al. 1998; Durfee et al. 1989, Sycara 1989). These strategies can be assigned to solve different kinds of conflicts. However, it is often not enough to decide which strategy to use solely based on the type of conflict since the same kind of conflict might require different resolution strategies depending on the particular objects that are involved. Hence, it is necessary to support variable conflict resolution strategies also at the object level.

Flexibility is a necessary feature of conflict resolution that can be challenging to implement and support. In adaptive hypermedia systems, authors can get annoyed if they are required to specify the adaptation strategy for every object. Hence, systems need to provide a default strategy for objects and types of conflicts such that authors and users can get the basic functionality without being required to invest excessive amounts of time and effort. However, default strategies are not always appropriate and the system must allow authors and users to override the default resolution strategies in order to meet their preferences.

The selection of strategies follows the idea used in CSS, where style rules cascade such that the most specific one is selected (Bos et al. 2005). In this regard MASH the least specific strategies are those specified by the system designers, followed by those specified by the authors and finally those specified by the readers.

4. WARP

WARP is a multi-model adaptive spatial hypermedia system written in Javascript and Java that executes inside IE and Netscape Web-browsers. By virtue of being a Web-based application, WARP circumvents some of the distribution issues typically associated with proprietary spatial hypermedia systems.

Figure 1 shows a WARP implementation of the abovementioned Financial News system running on Internet Explorer 6. Visible in the figure are objects representing news stories that the system has collected and presented to the user. Also visible in Figure 1 are two collections, each representing an industry included in the broker's portfolio ("Technology" and "Health"). These collections that contain the objects that represent the individual companies that the broker operates. This is the initial presentation of the information, it has not been adapted nor has the broker interacted with it yet.

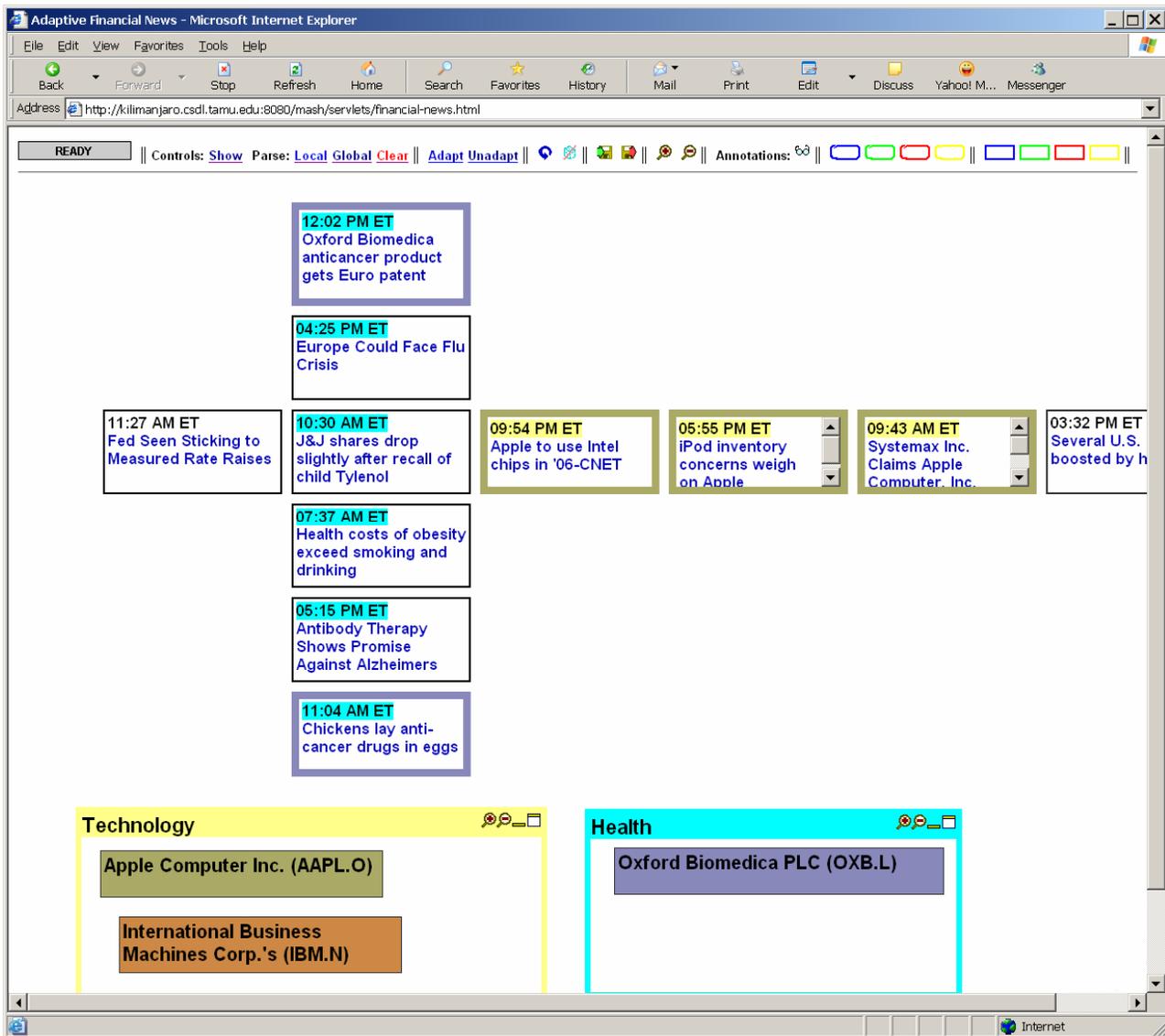


Figure 1. Financial News in WARP

On the top left corner of Figure 1 is WARP's control applet. This applet is in charge of parsing of the space and executing the adaptation process, while the visualization and interaction is handled by the Javascript components. This separation allows system designers to reuse the adaptation mechanisms in other systems.

The architecture of WARP is shown in Figure 2. It includes all the functional aspects of spatial hypermedia: generator, platform, spatial parser, analyzer and transformer. The spatial hypermedia generator supports the authoring of the spatial hypermedia. In WARP, this is accomplished in collaboration with VKB (Shipman et al. 2001). The platform refers to the system component that allows users to interact with the document. The spatial parser, as previously mentioned, is responsible for recognizing the implicit structures present in the document. The analyzer is in charge of evaluating the document based on the metrics available. The transformer is the component that computes the actual adaptations to the document.

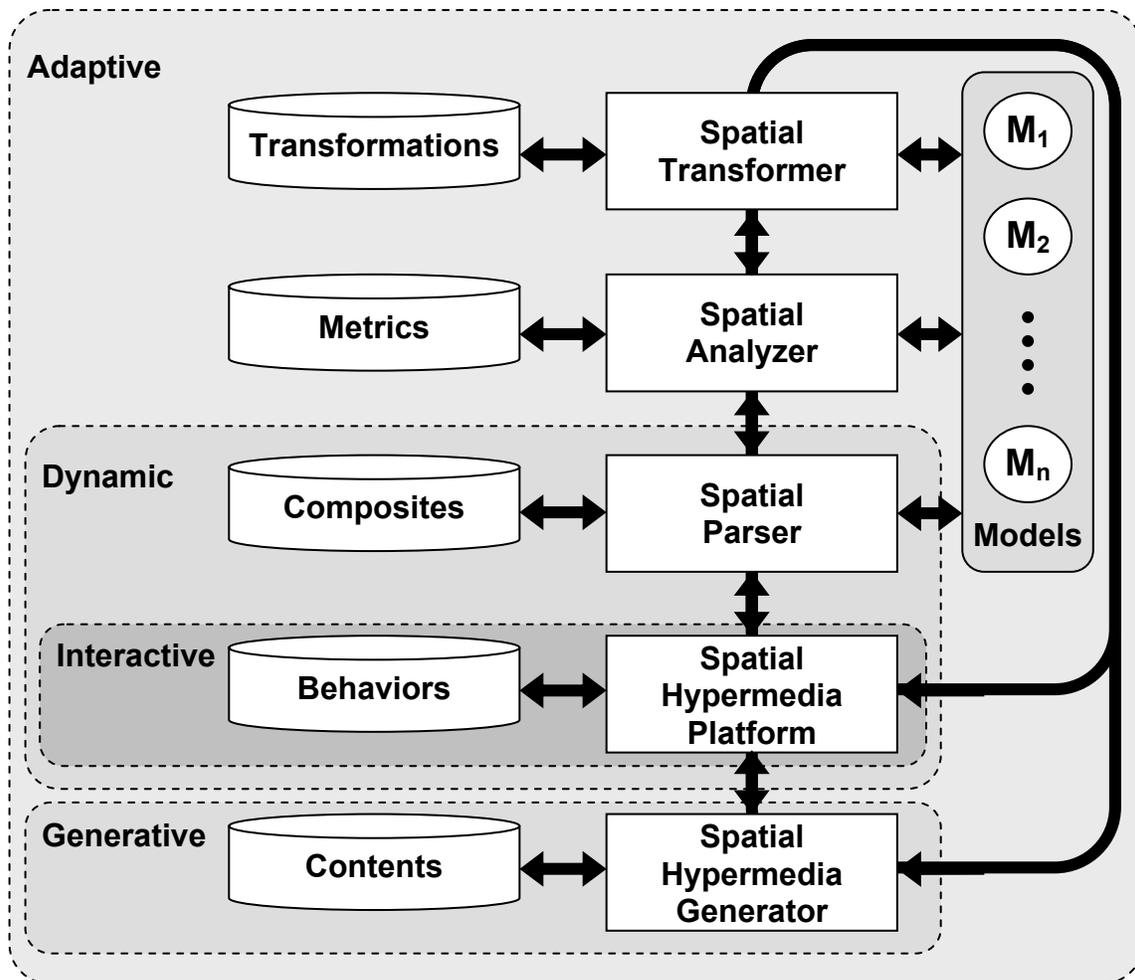


Figure 2. Spatial Hypermedia Framework

The adaptation process in WARP is carried out mainly by the spatial parser, analyzer and transformer and is guided by the set of available models. This process is explained in the following section.

4.1 Adaptation Process

WARP's approach to adaptation is designed to take an existing spatial hypermedia document and produce, guided by the set of models, an improved version of it.

The adaptation process is elaborate and can vary. It is best understood by tracing its functionality through an example. Figure 3-1 shows a sample presentation of the Financial News system before being adapted. The number of objects and complexity has been limited for clarity purposes.

At the spatial hypermedia platform, objects are simply located in space (Figure 3-1). There might be implicit relationships but the system has not yet inferred them. As previously mentioned, these implicit relationships define the context of the objects and recognizing them is critical for the resolution of conflicts. Hence, the first step in the process is to pass all objects to the spatial parser.

Figure 3-2 illustrates the results of parsing the document. The spatial parser has recognized a horizontal list, a vertical list and a pile composed of the horizontal and vertical lists. In this example, the spatial parser disregards piles of implicit composite objects.

Once the parser has finished, it passes the resulting object structure, comprised of all composite and atomic objects, to the spatial analyzer, as shown in Figure 3-3. At this point, the inferred structure is analyzed and the context of possible conflicts is inferred. This is the stage where conflict prevention techniques can be used. E.g., the analyzer can automatically instantiate behaviors that enforce a minimum distance between objects in order to prevent the possibility of unintentionally overlapping objects due to adaptations that move or resize the objects. The implementation of behaviors is discussed in section 4.3 in more detail

After deciding which conflict prevention strategies to use – guided by the suggestions from the models – the analyzer passes the objects and recognized structures to the transformer. At this point the transformer requests all models to provide their adaptation suggestions for all the objects as shown in Figure 3-4. These suggestions specify the method to be used, the strength of the method and the model's confidence on the suggestion itself.

Once all suggestions are collected, each object combines them using a conflict resolution strategy. The strategy employed can be the default strategy or an alternative strategy explicitly specified by the author or reader. WARP's set of conflict resolution strategies include:

- **Weighted average:** averages the strength of the suggestions, weighted by the object's confidence in the model and the model's confidence in the suggestion.
- **Maximum strength:** uses the suggestion with the highest strength.
- **Minimum strength:** uses the suggestion with the lowest strength.

- Maximum confidence in model: uses the strength suggested by the model that the object trusts the most. If the object trusts two or more models with the same maximum confidence, then the suggestions from these models are weight-averaged.
- Maximum suggestion confidence: uses the suggestion from the model with the highest confidence in the suggestion. If more than one model trusts its suggestion with the same maximum confidence, their suggestions are weight-averaged.
- Heuristic best: uses the suggestion with the highest heuristic value. The heuristic value is the product of the object's confidence in the model multiplied by model's confidence in the suggestion.

When using resolution strategies that select suggestions based on priorities such as the strength magnitude value or confidence value, deadlocks can occur when multiple suggestions have equal values. E.g., imagine WARP is using the maximum strength resolution strategy and among all suggestions two have the same maximum strength magnitude value. In these cases, WARP resolves the tie by computing the weighted average of those suggestions. Since suggestions can always be averaged, this approach guarantees reaching a final resolution.

Once each object has combined the suggestions from every model, the transformer then translates the high-level adaptation methods suggested into actual adaptation techniques. This translation is accomplished by revising the available transformations. Transformations are mappings that define which adaptation techniques can be used to implement every adaptation method. Transformations operate at an abstract level. The actual visualization occurs when the adaptation process is completed and the platform receives the final results.

Figure 3-5 shows the objects after registering all the suggestions (the composite objects have also received suggestions from the models, but they are not shown for clarity purposes). The object in the center has integrated all its adaptation suggestions and has concluded that it is very risky and needs to be protected from viewing. The object on the left has consolidated its suggestions resulting in a higher relevance that requires the object to be emphasized. The default mapping from adaptation methods to specific adaptation techniques can vary across objects. In the case of the left object, the default is to increase the border width. The default mapping for preventing the central object from viewing is to hide it.

The default mappings of adaptation methods may, or may not, produce the best results. Hence, the transformer requests the analyzer to evaluate the results of the transformations, as shown in Figure 3-6.

Using the context provided by the inferred spatial structure, the analyzer detects possible conflicts including: parent conflicts, parent-child conflicts and parent-parent conflicts. In addition, the analyzer identifies the set of constraints that need to be met when mapping the adaptation methods to techniques. For instance, in Figure 3-6, the analyzer has detected a parent-parent conflict – hiding the central object has broken the visual appearance of both lists. In order to search for better alternatives, the analyzer returns the objects to the transformer along with the identified mapping constraint to not hide the central object.

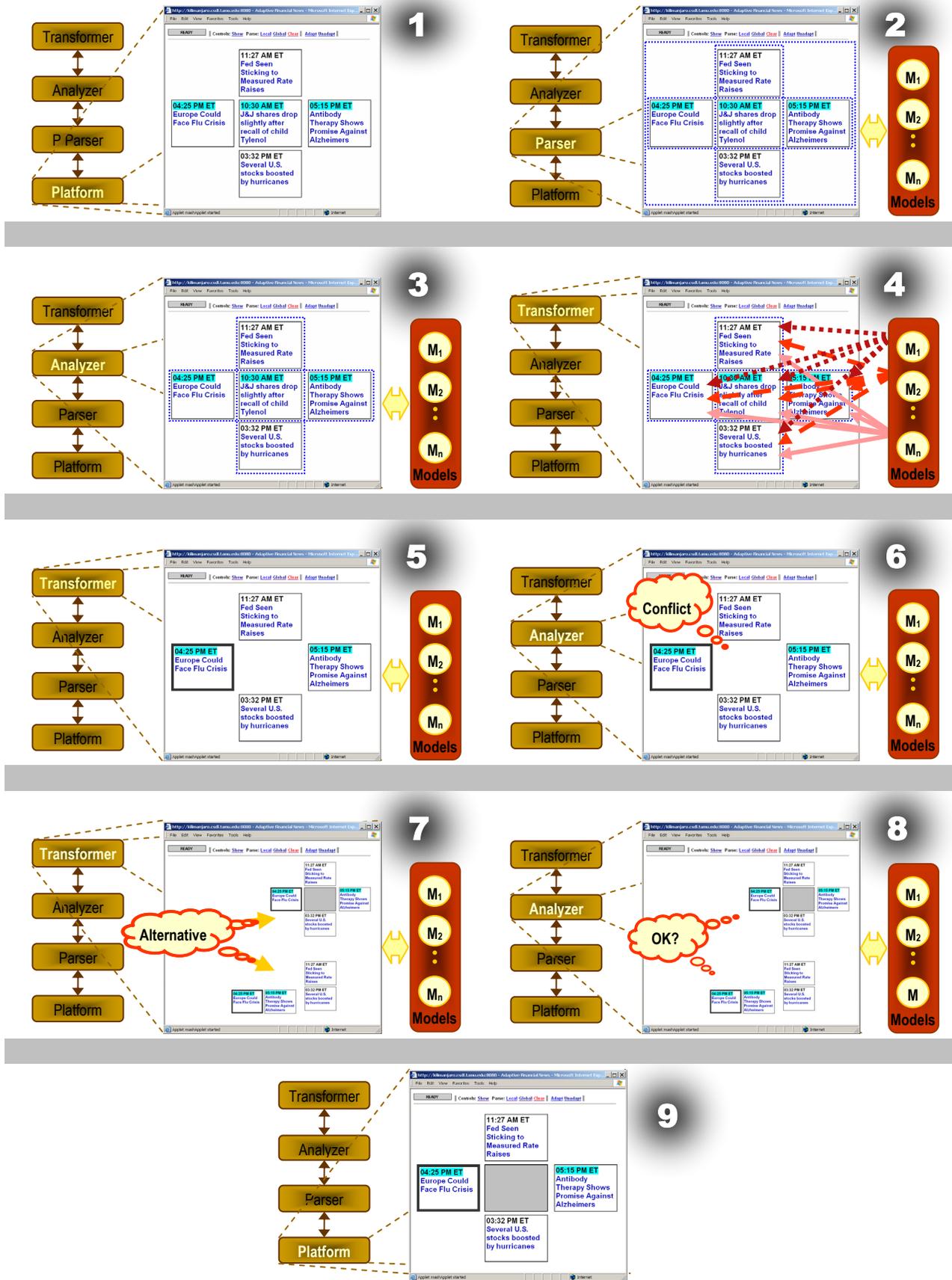


Figure 3. Adaptation Process

The transformer then needs to revise the available transformations to select alternative mappings that meet the constraints set by the analyzer. This can produce different configurations. Figure 3-7 shows two alternatives for the “protect from viewing” method.

One alternative – shown at the top of Figures 3-7 and 3-8 – is to gray-out the central object. This prevents content from being viewed but maintains the presence of the object in the space. This new adaptation is sent back to the analyzer. The analyzer then evaluates it. The adaptation is accepted if the analyzer does not detect conflicts. But if new conflicts are detected then the process iterates, requesting the transformer to generate another alternative adaptation. For instance, while normally the grayed-out alternative would have been accepted, for the purpose of this discussion assume momentarily that it was not. In this scenario, the transformer then can ask both parent composites (the vertical and horizontal lists) to react to the hiding of one of their members. As a result, both of them “contract”, moving their visible members together. The resulting structure is then passed to the analyzer, which then detects a new conflict and returns it to the transformer along with the new constraint to avoid overlapping objects. The transformer then moves the lists in order to meet this constraint and returns it to the analyzer. This result is the second alternative shown at the bottom of Figures 3-7 and 3-8.

Once the transformation alternative is deemed appropriate, or the transformer is incapable of finding better alternatives, the analyzer passes the adapted objects to the parser, which in turn passes them to the spatial hypermedia platform.

The last stage of the adaptation process occurs at the spatial hypermedia platform where the adaptation techniques are instantiated and the user is presented with the adapted document, as illustrated in Figure 3-9.

4.2 Conflicts and Constraints

A constraint describes a relationship that must or should be maintained (Freeman-Benson et al 1990). The behaviors instantiated in WARP as part of the adaptation process as described in section 4.1 (e.g., fixing the distance between two objects) represent examples of constraints. Constraint-based systems have been used successfully in user interfaces (Borning and Duisberg 1987; Hosobe 2001), graphical layout (Ryall et al. 1997; Sutherland 1963), information visualization, simulation (Borning and Duisberg 1987), and general purpose programming languages. They provide a convenient way to represent and automatically maintain relationships between objects. However, given that different constraints can interact with each other, finding a solution that satisfies all constraints can be very difficult. The research in this area is extensive and many approaches have been devised such as constraint hierarchies (Freeman-Benson et al 1990), force-directed algorithms (Tamassia 1998), and constraint solvers (Freeman-Benson et al 1990).

Constraints are of great relevance in MASH and more specifically in WARP, since its adaptation process relies on them to:

1. Prevent conflicts by explicitly enforcing the graphical relationships between objects
2. Detect conflicts by providing explicit ways to identify the context of conflicts
3. Resolve conflicts appropriately by identifying the possible propagation paths for conflicts

However, constraints also require a constraint solver. Currently, WARP uses a simplistic implementation of a constraint solver. The objective was to prove the feasibility of the overall conflict management approach. WARP's constraint solver cannot deal with cycles and performs a complete search for the best solution, resulting in a time complexity that limits its ability to handle large sets of constraints. Future work is necessary to identify improvements to the current solver.

One alternative to improve the constraint solver is to explore the use of previously developed approaches. However, since the characteristics of the constraints often limit the approach used for the constraint solver, it can be challenging to find an efficient approach capable of handling all the possible kinds of constraints in WARP. This challenge originates in the MASH approach, which by design allows document authors to create their own constraints without setting limits on the kinds of constraints that authors can define.

However, there are other features of MASH that can facilitate the facilitate finding a solution that satisfies the constraints. In MASH, the constraints can be transient, created and destroyed by the system as part of the adaptation process. This limits the computational overhead of the constraint solver to the adaptation process and minimizes its effects on the normal user interactions. Since the system-created constraints are instantiated in order to represent and enforce the implicit relationships inferred by the spatial parser, an alternative approach is to develop search heuristics based on the characteristics of the visual structures – e.g., piles, labeled lists, etc. This research, currently underway, attempts to find algorithms that take advantage of specific properties of the structures and solves them locally in order to limit the search space. It also attempts to determine whether the kinds of constraints currently being used are the most appropriate or if there are other kinds that can provide satisfactory solutions and at the same time are easier to solve.

4.3 Key Implementation Features

The set of models used by WARP can be fixed and predetermined by the author or can be variable and partially determined by the reader. In the first case, authors specify the set of models to be used and include them in the document as part of the adaptation applet. Alternatively, authors can specify the use of external models that can be located somewhere else on Internet. By using a naming scheme, WARP can resolve the location and identity of the models at run time. This enables authors to specify the use of models provided by the reader. In this scheme, readers manage and maintain their own personal user models, releasing authors from creating and managing individual profiles. Readers benefit by being able to reuse their models in different domains and applications. Since this approach bypasses the need to send sensitive information over the Internet, readers also benefit by keeping their private information safely in their own machine.

WARP supports specifying whether models are used for the adaptation of the whole document or just individual objects. While by default objects accept suggestions from all models available, each object has a list of the models from which it accepts adaptation suggestions. Objects also maintain a confidence assessment for each model. This confidence value represents how much the object trusts the model. This way authors can configure objects to only follow – or at least prefer – the adaptation suggestions of models specifically designed for a particular type of object.

The only constraint necessary for any independent model to participate in the adaptation process in WARP is that it suggests adaptations at a very high level using adaptation methods known to the system. Figure 4 shows WARP’s adaptation suggestions API.

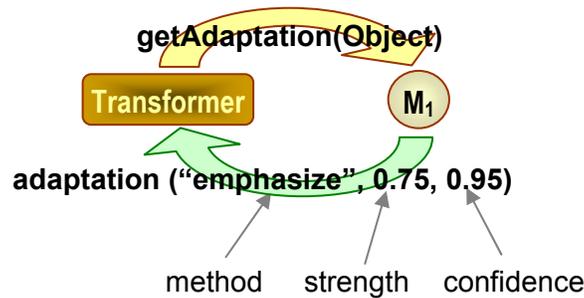


Figure 4. Adaptation Suggestion

As illustrated in Figure 4, the models’ adaptation suggestions have three values, an adaptation method, a strength magnitude value (ranging from -1 to 1) and the model’s confidence of that assessment (ranging from 0 to 1). The adaptation method needs to be known to the transformer such that it can be instantiated into an actual adaptation technique. The strength and confidence values, together with the value of the objects’ confidence in the models, are necessary to enable the transformer to apply its conflict resolution strategies. WARP allows any models that produces properly formed suggestions to be reused for any spatial hypermedia document.

4.4 Approach Evaluation

The MASH approach to adaptation supported by WARP has gone through a first round of evaluation (Francisco-Revilla 2005; Francisco-Revilla and Shipman 2004a). This study evaluated the efficacy of the adaptation approach in meeting the goals of two independent models whose adaptation suggestions conflicted. The study showed that the automatic resolution of conflicts produced effective adaptations that augmented the effectiveness of the spatial hypermedia document as an instructional medium. The adapted document facilitated the users’ assigned task of authoring a Web page when compared to a non-adapted version of the same document. While this study validated of the overall adaptation approach, it tested only a subset of the possible conflict management techniques supported by the system. A thorough evaluation of all the features is scheduled to be conducted.

5. CONCLUSIONS

New adaptive systems already employ multiple models to guide their adaptation process. However, in order to improve the systems responsiveness, it is necessary to support of a *variable set of independent* models.

However, this approach is challenging as conflicts can occur between the suggestions from different models. In adaptive spatial hypermedia the challenge is even greater since conflicts can propagate across relationships between objects. This requires of a comprehensive conflict management scheme that encompasses the processes of conflict prevention, conflict detection and conflict resolution.

WARP is a Web-based system that illustrates how multi-model adaptive spatial hypermedia systems can deliver presentation-oriented spatial hypermedia documents over the Web and adapt them using a variable set of multiple independent models. WARP provides a comprehensive conflict management approach that exploits the features of spatial hypermedia. By using high-level adaptation suggestions WARP provides coherent adaptive behavior even when employing independent models that are not explicitly coordinated.

WARP illustrates how conflicts can be prevented by taking advantage of spatial hypermedia's support of ambiguity and by dynamically selecting the appropriate mappings from adaptation methods to techniques such that the adaptations can be simultaneously represented. WARP's spatial parser facilitates the detection of conflicts by determining their context and scope. This enables WARP to use the appropriate conflict resolution strategy before the conflicts propagate through the object relationships.

6. ACKNOWLEDGMENTS

This work is partially funded by NSF grants. IIS-9734167, DUE-0121527 and DUE-0226321.

7. REFERENCES

- Borning, A., and Duisberg, R.** (1986) Constraint-Based Tools for Building User Interfaces. In *ACM Transactions on Graphics* 5, 4, October (New York, ACM Press), pp. 345-374.
- Bos, B., Lie, H., Lilley, C., and Jacobs, I.** (2005) *Cascading Style Sheets, Level 2 Revision 1, CSS 2.1 Specification*. W3C Working Draft, January. <http://www.w3.org/TR/CSS21/>
- Brusilovsky, P.** (1996) Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction, Vol. 6 No.2-3* (Dordrecht, Kluwer Academic Publishers), pp. 87-129.
- Brusilovsky, P. and Cooper, D. W.** (2002) Domain, Task, and User Models for an Adaptive Hypermedia Performance Support System In *Proceedings of the International Conference on Intelligent User Interfaces '02*, San Francisco, California, USA - January 13-16, (New York, ACM Press), pp. 23-30.
- Buchanan G., Blanford, A., and Jones, M.** (2004) Integrating Information Seeking and Structuring: Exploring the Role of Spatial Hypertext in a Digital Library. In *Proceedings of the Fifteenth ACM Conference on Hypertext & Hypermedia*, (New York, ACM Press), pp. 225-234.

- Conry, S., Meyer, R.A. and Lesser, V.R.** (1998) Multistage Negotiation in Distributed Artificial Intelligence. In Bond, A. and Gasser, L. (Ed.) *Readings in Distributed Artificial Intelligence*. (San Mateo, CA, Morgan Kaufman).
- De Bra, P., Houben, G-J., and Wu, H.** (1999) AHAM: a Dexter-based reference model for adaptive hypermedia. In *Proceedings of the 10th ACM Conference on Hypertext and Hypermedia: Returning to Our Diverse Roots*, Darmstadt, Germany, February 21-25, (New York, ACM Press), pp. 147-156.
- Durfee, E.H., Lesser, V.R., and Corkill, D.D.** (1989) Cooperative Distributed Problem Solving. In Barr, A., Cohen, P. R., and Feigenbaum, E. A., (Ed.). *The Handbook of Artificial Intelligence*. Vol. IV. (Reading, MA, Addison Wesley), pp. 83-147.
- Encarnação, L.M.** (1997) Multi-Level User Support through Adaptive Hypermedia: A Highly Application-Independent Help Component. In *Proceedings of Intelligent User Interfaces '97*, Orlando, Florida - January 6-9, (New York, ACM Press), pp. 187-194.
- Ferber, J.** (1999) *Multi-Agent Systems. An Introduction to Distributed Artificial Intelligence*. (Reading, MA, Addison-Wesley).
- Francisco-Revilla L.** (2005) *Multi-model Adaptive Spatial Hypermedia*. Ph.D. Dissertation, (College Station, TX. Texas A&M University).
- Francisco-Revilla L.** (1998) *User & Situation Models for Medical Information Delivery*. Masters Thesis. Dept. of Computer Science. (College Station, TX. Texas A&M University).
- Francisco-Revilla L., and Shipman, F.M.** (2005) Parsing and Interpreting Ambiguous Structures in Spatial Hypermedia. To be published in the Sixteenth ACM Conference on Hypertext and Hypermedia HT'05, Salzburg, Austria, September 6-9, 2005, (New York, ACM Press), pp. TBA.
- Francisco-Revilla L., and Shipman, F.M.** (2004a) Instructional Information in Adaptive Spatial Hypertext. In *Proceedings of Document Engineering 2004*, (New York, ACM Press), pp. 124-133.
- Francisco-Revilla L., and Shipman, F.M.** (2004b) Managing Conflict in Multi-model Adaptive Hypertext. In *Proceedings of the Fifteenth ACM Conference on Hypertext & Hypermedia* (New York, ACM Press), pp. 237-238.
- Francisco-Revilla L., and Shipman, F.M.** (2004c) WARP: a Web-based Dynamic Spatial Hypertext. In *Proceedings of the Fifteenth ACM Conference on Hypertext & Hypermedia*, (New York, ACM Press), pp. 235-236.

- Francisco-Revilla L., and Shipman, F.M.** (2000) Adaptive Medical Information Delivery: Combining User, Task and Situation Models, in *Proceeding of IUI 2000*, New Orleans, USA, Jan 9-12, (New York, ACM Press), pp. 94-97.
- Freeman-Benson, B. N., Maloney, J., and Borning, A.** (1990) An Incremental Constraint Solver. *Communications of ACM* 33, 1, January (New York, ACM Press), pp. 54-63.
- Grønbaeck, K, Vestergaard, P., P., and Ørbæk, P.** (2002) Towards Geo-Spatial hypermedia: Concepts and Prototype Implementation. In *Proceedings of HT 2002*, Maryland, USA, June 11-15, (New York, ACM Press), pp. 117-126.
- Hosobe, H.** (2001) A Modular Geometric Constraint Solver for User Interface Applications. In *Proceedings of the 14th Annual ACM Symposium on User interface Software and Technology*, Orlando, Florida, November 11 - 14 (New York, ACM Press), pp. 91-100.
- Hsieh, H. and Shipman, F.** (2000) VITE: A Visual Interface Supporting the Direct Manipulation of Structured Data Using Two-Way Mappings. In *Proceedings of IUI 2000*, (New York, ACM Press), pp. 141-148.
- Malone, T.W.** (1983) How do People Organise their Desks? Implications for the Design of Office Information Systems. In *ACM Transactions on Information Systems* 1:1 (January 1983), ACM Press, pp. 99-112.
- Marshall C.C., Halasz, F.G., Rogers, R.A., and Janssen W. C. Jr.** (1991) Aquanet: A Hypertext Tool to Hold Your Knowledge in Place. In *Proceedings of the Third ACM Conference on Hypertext*, (New York, ACM Press), pp. 261-275.
- Marshall C.C., and Rogers R.A.** (1992) Two Years before the Mist: Experiences with Aquanet, In *Proceedings of HT'92*, Milano, Italy, November 30-December 4, (New York, ACM Press), pp. 53-62.
- Marshall C.C., and Shipman, F.M.** (1993) Searching for the Missing Link: Discovering Implicit Structure in Spatial Hypertext. In *Proceedings of 5th ACM Conference on Hypertext*, (New York, ACM Press), pp. 217-230.
- Marshall C.C., Shipman, F.M., and Coombs, J.H.** (1994) VIKI: Spatial Hypertext Supporting Emergent Structure. In *Proceedings of the 6th ACM European Conference on Hypermedia Technology*, (New York, ACM Press), pp. 13-23.
- Nakakoji, K., Yamamoto, Y., Takada, S., and Reeves, B.N.** (2000) Two-Dimensional Spatial Positioning as a Means for Reflection in Design. In *Proceedings of the Conference on Designing Interactive Systems* (New York, ACM Press), pp. 145-153.
- Ryall, K., Marks, J., and Shieber, S.** (1997) An Interactive Constraint-Based System for Drawing Graphs. In *Proceedings of the 10th Annual ACM Symposium on User interface Software and Technology*, Banff, Alberta, Canada, October 14 - 17 (New York, ACM Press), pp. 97-104.

Sandholm, T.W. (1999) Distributed Rational Decision Making. In Weiss, G. *Multi-Agent Systems*. Pp. 201-258. (Cambridge, MA, MIT Press).

schraefel, m.c. (2000) ConTexts: Adaptable Hypermedia. In *Proceedings of AH 2000*, Trento, Italy, August, (New York, Springer-Verlag), Lecture Notes in Computer Science, Vol. 1892, 369-375.

Shipman, F.M., Marshall, C.C., and Moran, T.P. (1995) Finding and Using Implicit Structure in Human Organized Spatial Layouts of Information. In *Proceedings of CHI'95*, Denver, Colorado, USA, May 7-11, (New York, ACM Press), pp. 346-353.

Shipman, F., Hsieh, H., and Airhart, R., Maloor, P., and Moore, J.M. (2001) The Visual Knowledge Builder: A Second Generation Spatial Hypertext. In *Proceedings of Hypertext 2001*, (New York, ACM Press), pp. 113-122.

Shipman, F.M., and McCall, R. (1994) Supporting Knowledge-Base Evolution with Incremental Formalization. In *Proceedings of CHI'94*, Boston, Massachusetts, USA, April 24-28, (New York, ACM Press), pp. 285-291.

Stephanidis, C., Karagiannidis, C., and Koumpis, A. (1997) Decision Making in Intelligent User Interfaces. In *Proceedings of IUI'97*, Orlando, Florida - January 6-9, (New York, ACM Press), pp. 195-202.

Suchman, L.A. (1987) Plans and Situated Actions. (New York, Cambridge University Press).

Sutherland, I.E. (1963) Sketchpad: A Man-Machine Graphical Communication System. In *Proceedings of the Spring Joint Computer Conference*, IFIPS, pp. 329-346

Sycara, K. (1989) Multiagent Compromise via Negotiation. In Gasser, L. and Huhns, M. *Distributed Artificial intelligence*. (San Mateo, CA: Morgan Kaufmann).

Tamassia, R. (1998) Constraints in Graph Drawing Algorithms. *Constraints* 3, 1 (April), pp. 87-120.

Trigg R.H., and Weiser, M. (1986) TEXTNET: A Networking-Based Approach to Text Handling. In *ACM Transactions on Office Information Systems*, 4, 1, January (New York, ACM Press), pp. 1-23.

Web Squirrel:

<http://www.eastgate.com/squirrel/FAQ.html>