Information Assembly for Adaptive Display

Andrew LAMPERT
CSIRO ICT Centre
Locked Bag 17
North Ryde, Australia, 1670
Andrew.Lampert@csiro.au

Cécile PARIS
CSIRO ICT Centre
Locked Bag 17
North Ryde, Australia, 1670
Cecile.Paris@csiro.au

Abstract

In a world where information is increasingly delivered to users via different devices with dramatically different constraints and capabilities, it is becoming crucial to consider how the presentation of information must be adapted to suit specific devices and user contexts. To avoid confusing and disorienting the users as they switch between devices, the content and structure of information should be kept constant while its presentation must be optimised for each device to ensure usability. In this paper, we distinguish between two types of decisions that must be made during the presentation planning stage of an information delivery system: local decisions, which are based only on the content or features of the node itself, and global decisions, which are based on the entire structure of the discourse tree. We present a generic algorithm for making global decisions, driven by the discourse tree structure and contextual characteristics.

1 Introduction

In a world where information is increasingly delivered to users via different devices with dramatically different constraints and capabilities, it is becoming crucial to consider how the presentation of information must be adapted to suit specific devices and user contexts. To perform this adaptation and customisation manually is an expensive and time-consuming task. Additionally, as devices become increasingly interconnected, and as users are able to switch from one to another at will, it is essential to avoid confusing and disorienting the user (Chincholle, 2000). To this end, it is desirable to keep the content and structure of a document constant while optimising the presentation for each device to ensure usability. This is especially true for small or mobile devices.

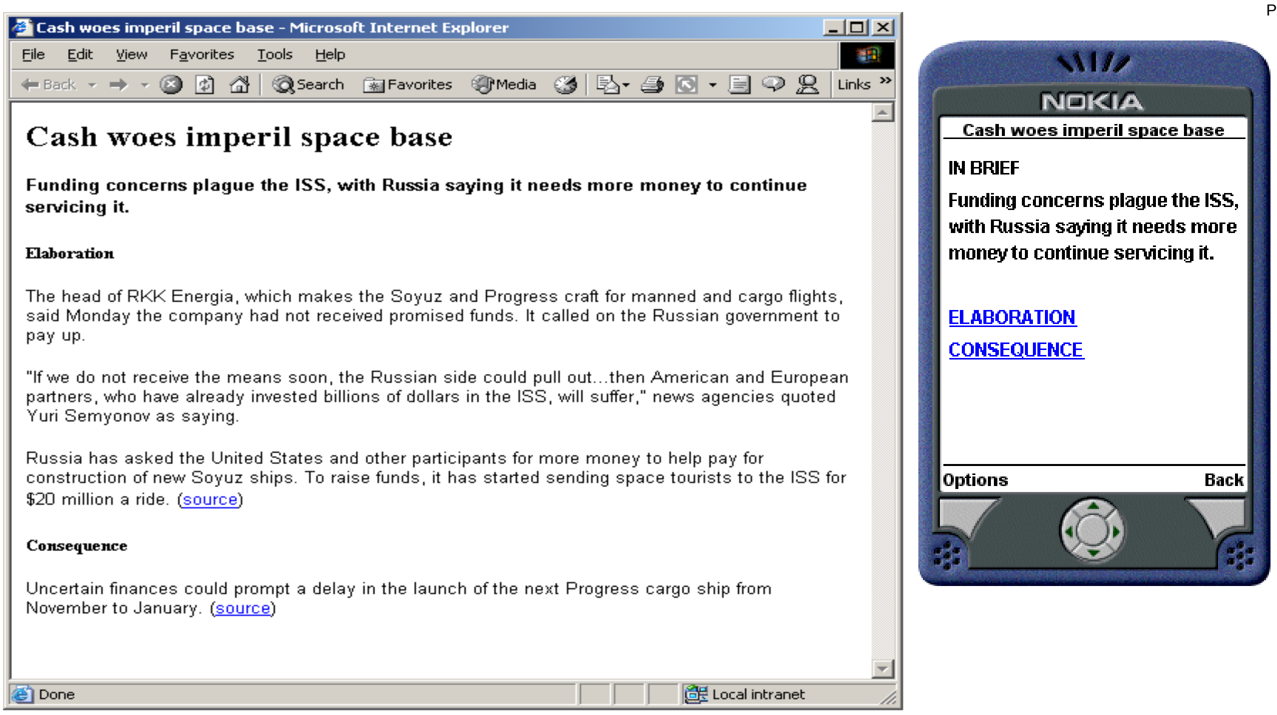
Figure 1 illustrates this issue. The screen on the left shows a report that might have been dynamically generated for a PC based web browser in response to a query about "financial threats to the Russian space program". This report integrates information retrieved from multiple sources, and the information is structured and organised in order to be easily understandable. In particular, each piece of information is labelled with its role in the report. The most important part of the report is shown in bold at the very top. Then there is *elaboration* information, which gives more information about this first sentence. This is followed with information indicating the likely impact (*consequence*). The right hand side of Figure 1 shows the presentation of the same information, now delivered on a mobile phone. Given the different affordability of the delivery channel, the presentation and, in particular, the navigation is quite different, while the content and structure of the information being presented is constant.

Our approach to ensuring coherence and consistency of information across devices is to use Myriad (Paris *et al.*, 2004), a platform for contextualised information retrieval and delivery based on theories and techniques from natural language generation. Our system produces virtual documents (or, more generally, presentations) by essentially going through two main stages: first it selects, retrieves and organises information to present to the user, taking the context into account (e.g., the task at hand, the user, the environment); then, taking the result of the first stage, it decides how to best deliver the selected information, once again taking the context into account, in particular the delivery device.

The remainder of this paper is structured as follows: we first present the Myriad architecture and briefly explain how it allows for coherent tailored information delivery. We introduce the two main stages involved: content and structure planning, and presentation planning. We then explain in detail the presentation planning stage, presenting how we came to decide that this stage was itself divided into two steps, which we present in turn, focusing especially on the "information assembly" step.

Proceedings of the Australasian Language Technology Workshop 2004
Macquarie University, Sydney, December 8th, 2004. Copyright, Australian Speech Science & Technology Association Inc.

Accepted after full paper review


Cash woes imperil space base - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media Links
Done
Local intranet
NOKIA
Options
Back

Cash woes imperil space base

Funding concerns plague the ISS, with Russia saying it needs more money to continue servicing it.

Elaboration

The head of RKK Energia, which makes the Soyuz and Progress craft for manned and cargo flights, said Monday the company had not received promised funds. It called on the Russian government to pay up.

"If we do not receive the means soon, the Russian side could pull out...then American and European partners, who have already invested billions of dollars in the ISS, will suffer," news agencies quoted Yuri Semyonov as saying.

Russia has asked the United States and other participants for more money to help pay for construction of new Soyuz ships. To raise funds, it has started sending space tourists to the ISS for $20 million a ride. (source)

Consequence

Uncertain finances could prompt a delay in the launch of the next Progress cargo ship from November to January. (source)

Cash woes imperil space base

IN BRIEF

Funding concerns plague the ISS, with Russia saying it needs more money to continue servicing it.

ELABORATION

CONSEQUENCE

Figure 1: Adapting the presentation of information to suit different devices

Finally, we briefly introduce the last stage of the information delivery process, the realisation. The paper ends with some concluding remarks.

2 The Myriad Architecture

The core of the Myriad architecture is our information planning engine, which we call the Virtual Document Planner (VDP) (Colineau *et al.*, 2004b). The VDP is based on a typical Natural Language Generation (NLG) architecture, where the linguistic resources are separate from the planning engine. The VDP is based on the Moore and Paris (1993) text planner, and, as in that text planner, the resources are represented as plans.

The VDP works essentially as follows. Given a top level communicative goal (an overall purpose for presenting information), the engine uses a library of discourse rules (plan operators) to select and organise the content. Then, a library of presentation plans is employed to make local presentation decisions about how to present content. The output of the discourse planning stage is a discourse tree, which is then augmented (extended) during the presentation planning stage. This is shown schematically in Figure 2, where *content* represents the tree constructed as a result of the content and structure planning stage, and *presentation* represents the extension done during the presentation planning stage to take the specific delivery device into account. As we see from the figure, content and structure can thus remain constant across devices. Finally, the content represented in the extended tree is realised in syntax appropriate for the output device during the realisation phase. For example, referring back to Figure 1, during this phase, the content tree would be realised into HTML for the PC based display, while the mobile delivery might require WML.

As in (Moore and Paris, 1993), our approach exploits rhetorical relations, also called coherence relations, based on Rhetorical Structure Theory (RST) (Mann and Thompson, 1988) to guarantee the coherence of the resulting presentation. The discourse and presentation rules both specify coherence relations that must hold between sibling sub-goals created by each goal decomposition. These coherence relations indicate how the various discourse segments and pieces of information work together to achieve the top level communicative goal. This was illustrated in Figure 1, where the second paragraph was related to the first paragraph by an *elaboration* relation, while the third paragraph was related to it by a *consequence* relation. Using terminology from RST, the first paragraph is the *nucleus*, while the other two paragraphs are *satellites*.

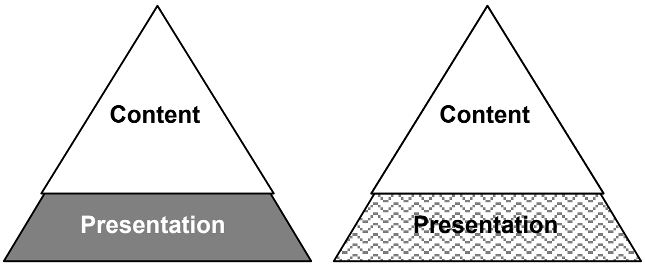


Figure 2: Content and presentation planning for two different devices

For reasons of simplicity and modularity, we explicitly maintain a conceptual and architectural separation between the different processes of content planning, presentation planning and surface realisation. This separation is closely analogous to the distinction between document planning, micro planning and surface realisation adopted by many natural language generation systems (e.g., McKeown, 1985; Hovy 1988; Moore and Paris, 1993).

We exploited this generation paradigm to build several prototypes of information delivery systems, in particular one in the travel domain (cf., Wilkinson *et al.,* 2000; Paris *et al.*, 2001; Paris, 2002) and one in the corporate domain, where users received a brochure about CSIRO tailored to their interest and needs (cf. Paris *et al.*, 2003). In these prototypes, the process was exactly as illustrated in Figure 2.

Through these prototypes, however, we have now come to recognise that, during the presentation planning stage, we must distinguish between two types of decisions:

- *local* decisions, which are based only on the content or features of the *node* itself, and
- *global* decisions based on the *entire structure* of the discourse tree.

An example of a local decision might be to ‘present the content of this node in a table’ or to present that content as a bulleted list. In contrast, deciding to create a navigation index and organising content pages appropriately around this index requires more global knowledge of the discourse tree structure.

Thus, the process we illustrated in Figure 2 can really only be used to make local decisions. Failing to make the distinction between these two types of decisions resulted in discourse operators that embedded decisions that should really occur during the presentation stage, while taking the whole discourse tree into account. This thus blurred the separation between content and presentation planning that we originally desired.

To address these shortcomings, we have designed a new layer within the presentation planning stage. This layer implements the reasoning that must occur during presentation planning but that must make decisions based on the global discourse structure. Our aim in the work presented here was four fold:

- keep the conceptual separation between content and structure planning and presentation planning, while still being able to make a number of decisions that affect the final presentation (in particular, decisions about navigation);
- explicitly decouple, in the presentation planning, local decisions from global decision;
- decouple the decision process (or algorithm) from the information needed to make decisions. This information might include characteristics of the device or of the user; and, finally,
- produce a generic set of operators which can be used to produce reasonable (but not necessarily highly optimised) output for a large range of information domains, discourse structures and output devices. This is in contrast with our other work, including the DFDMSA[1] project, where our aim is to produce multimedia output that is optimised for the task at hand (Colineau *et al.*, 2004).

In the remainder of this paper, after briefly describing the process implementing the local presentation decisions, we focus on the processing stage which enables the system to make global presentation decisions based on the discourse tree.

3 Information Presentation

We now propose to account for the two types of decisions we introduced in the previous section by having two steps, which we term “local presentation planning” and “information assembly”. Each of these steps exploits its own mechanism. We describe them in turn. As mentioned above, when the system reaches the presentation planning stage, a discourse tree has already been constructed through discourse planning.

3.1 Local Presentation Decisions

Local presentation planning is as we had originally envisioned and implemented it. This stage thus extends the current discourse tree. It is performed using standard Moore and Paris (1993) inspired plan operators.

[1] Data Fusion, Data Management and Situation Assessment.

```
<operator>
     <effect>InitialiseAssembly ?nodeId</effect>
     <constraint>(discourse:isRootNode(?nodeId))</constraint>
     <constraint>(set ?user (user:getCurrentUser()))</constraint>
     <constraint>(set ?deviceModel (device:getCurrentDevice(?user)))</constraint>
     <constraint>(set ?Layout (device:chooseLayout(?deviceModel)))</constraint>
     <constraint>
     (set ?contentFrame (layout:chooseContentFrame(?Layout ?nodeId)) )
     </constraint>
     <operation>
     (discourse:annotateFeature(contentPaneSpace device:getSpace(?contentFrame))
     </operation>
     <operation>(discourse:annotateFeature(contentFrame ?contentFrame))</operation>
     <operation>(discourse:annotateFeature(Layout ?Layout))</operation>
     <operation>(post ProcessRootNode ?Layout ?contentFrame)</operation>
</operator>
<operator>
     <effect>ProcessRootNode ?Layout ?contentFrame</effect>
     <constraint>(layout:hasIndexFrame(?Layout))</constraint>
     <constraint>(layout:isRefillable(?contentFrame))</constraint>
     <constraint>(set ?indexFrame (layout:getIndexFrame(?Layout)))</constraint>
     <constraint>(set ?depthLimit (layout:getDepthLimit ?indexFrame))</constraint>
     <constraint>(set ?realisationOrder 1)</constraint>
     <operation>(discourse:annotateFeature(indexDepth 0))</operation>
     <operation>(discourse:annotateFeature(indexFrame ?indexFrame))</operation>
     <operation>(discourse:annotateFeature(indexDepthLimit ?depthLimit))</operation>

     <operation>(foreach ?child (discourse:getChildrenInOrder(?nodeId))
        (post     ProcessNode     ?child     ?Layout     (layout:chooseContentPage(?child,
     ?contentFrame))
      ?realisationOrder++ ?indexFrame))
     </operation>
</operator>
```

Figure 3: Assembly operators for the root node of the discourse tree

As explained earlier, this stage augments the discourse tree at the leaf nodes by performing further goal decomposition. This stage uses a specific library of presentation plan operators. Decisions on how to present specific discourse content (contained in the leaf nodes of the discourse tree) are made during this step. Examples of presentation decisions made could include marking up titles with specific style and font information, or choosing to display certain information as a list. Throughout this process, presentation decisions are made based on node-local information; the global structure of discourse tree is not considered. In certain circumstances, presentation planning may reason about relations between immediate siblings, but not about more global tree structure. For example, if a node is related to a sibling nucleus by an RST relation “preparation”, a decision might be made to mark up the content of a node with specific style and font information appropriate for a heading. Previous work in natural language generation has shown that rhetorical text organisation, as embodied in the RST relations in a discourse tree, can be used to motivate text-formatting decisions (e.g., Hovy and Arens, 1991).

3.2 Global Presentation Decisions

The second stage deals with presentation decisions which require knowledge of the global structure of the discourse tree. This stage primarily *assembles* and *combines* pieces of information based on how they are related to each other within the discourse tree and on the constraints of the environment (e.g., the delivery device). We thus choose to refer to this stage as the i*nformation assembly* stage.

In this stage, the tree constructed thus far is traversed in a top-down fashion, as opposed to being extended. Each node is annotated with specifications that will be interpreted during the final realisation tree-walk. (Note that it is only during the final realisation pass through the discourse tree that the final output is actually produced.) This approach can be seen as an extension of the realisation process used in the DFDMSA project (Colineau *et al.* 2004).

For the sake of simplicity, information assembly is performed using another specific library of operators which implement the following algorithm.

Starting at the root node of the discourse tree, the VDP performs a top down pass through the tree, executing the following two functions at each node:

1. It annotates the current tree node with information required for correct realisation. These annotations are based on contextual characteristics. As mentioned before, the annotations are interpreted during final realisation.
2. It posts a new information assembly goal for each of its child nodes (if any). In this way, the discourse tree is traversed in a recursive manner, using the same sub goal decomposition mechanism of the existing plan operators and planning engine.

As we process the root node of the tree, a number of contextual models are accessed. In particular, information about the device used by the user is retrieved. This determines how to proceed. We have identified a number of attributes that may characterise the device. They include:

- the size of the screen (if there is a screen);
- styles of navigation;
- available modalities;
- software capabilities; and
- layout templates available for formatting on that specific device (e.g., stylesheets).

This stage of the information assembly is implemented by the operators shown in Figure 3. As illustrated in these operators, the contextual models are accessed through the *constraints* mechanism, allowing the system to choose appropriate operators based on these models. Note also the *operations* of the operators. It is through them that (1) the tree is augmented with annotations and (2) the system posts additional goals to continue the traversal of the tree.

Importantly, these operators are generic, and the decision process is decoupled from the information needed to make decisions, as was our goal. This information is encoded in declarative models outside the engine. These include the device model.

It is envisaged that the device model would eventually be encoded in a standard format such as that defined in the W3C's Composite Capabilities/Preferences Profile (CC/PP) framework (Klyne *et al.*, 2004). The RDF-based CC/PP framework provides a way to describe generic profiles which are accessible via the web, and could be created and maintained by hardware and software vendors for their own devices and applications. This mechanism would work well as a distributed contextual device model for the Myriad framework. Because the CC/PP framework has not yet been widely adopted, for now we are using a simpler XML-based representation for our device models.

Layout templates act as stylesheets to organize the presentation of information. Layout templates are not device specific, meaning that the same layout template can often be used across multiple devices. Equally, the same device may support a number of different layout templates for displaying content. Layout templates are characterised by properties such as:

- the set of frames contained;
- the content type, e.g., MIME content types (Freed and Borenstein, 1996), supported by each frame; and
- the relative size of each frame.

Once instantiated for a specific device, a layout template and its frames are additionally constrained by the capabilities of the device. For example, a device may determine the amount of content that can be displayed in each frame of a layout template, whether a frame can spawn new windows for displaying new content, and whether the frame supports scrollbars. All these features affect later decisions about how to allocate space to individual pieces of content, and how to display all the planned content if there is not enough space. Assembly operators reason about layout template characteristics to determine how the planned content will be presented in that template.

An example of a frame is an index frame. This contains a hierarchically indented set of links to content displayed in other frames. Index frames have additional attributes such as the maximum hierarchy depth supported for index entries.

If the chosen layout template includes an index frame, it will contain hyperlinked references to some of the content pages. Exactly which pages are able to add a reference to themselves in the index frame is controlled by the assembly operators. The maximum hierarchy depth of an index frame signifies the *indent depth limit* of the index. The assembly operators keep track of the number of content page references already inserted. At some point, a depth threshold will be reached (which represents a maximum level of indentation), and nodes will no longer be able to insert references into the index. At this point, all content for the discourse tree branch below that node must be delivered into a single, linear space (for most devices, this represents a single page instance). In this situation, an algorithm for allocating space is needed to divide up that linear space amongst the child nodes. Two possible algorithms are discussed below.

After initialisation is complete, the information assembly stage recursively processes the remaining nodes in the tree. This results in further annotations being made to the tree. Examples of features that may be annotated in each node include:

- ♦ The order in which a node should be realised in relation to its sibling nodes.

Importantly, this is determined by the discourse relations between each of the sibling nodes. For example, Figure 4 illustrates that, for the elaboration RST relation, the nucleus is realised before the satellite;

- Hyperlink anchor text;
- Whether a node should add a reference to itself in the navigation index (if there is one); and
- The exact page instance(s) to be used to realise the node's content.

Table 1 illustrates some specific examples of feature names and possible values, exactly as they are encoded in the discourse tree.

Feature Name	**Example Value**	**Description/ Interpretation**
realisationOrder	2	Realise this node second, after one sibling node.
addToIndex	True	Add a reference to this node's content in the navigation index frame
indexDepth	2	The depth of index entries at this node is 2.
indexText	Section Name	Use this text as the anchor in the index to link to this node's content.
anchorText	More Information	Use this text as the anchor text for this hypertext link.
linkTarget	Page001	The target of this hypertext link is the specified page.
contentPage	Page001	Realise the content of this node in the specified page.

Table 1: Examples of features that are annotated into the discourse tree during information assembly

The coherence relations are ranked in another declarative resource, so that authors can declare whatever subjective realisation ordering suits their specific purposes or information domain.

```
<relation>
  <name>elaboration</name>
  <type>RST</type>
  <description>The elaboration RST
        relation</description>
  <library>discourse</library>
  <importance>medium</importance>
  <order>N:S</order>
</relation>
```

Figure 4: Definition for the elaboration RST relation

Space Allocation within a single page

A layout template may not have a separate navigation index frame, as is often the case on small screen devices such as mobile phones or PDAs. Alternatively, the system may have descended through the discourse tree beyond the index depth threshold. At that point, there is a single linear piece of space in which to realise an entire segment of our presentation. Often, in such a situation, there will not be enough space to realise all the content directly. As a result, sections of content must be summarised or realised on separate pages that are hyperlinked into the single shared space.

We have identified two generic heuristics that could be used to allocate content to the available space. (This assumes we can characterise available space, e.g., by the number of lines of text that can be displayed). These heuristics are embodied in additional operators.

One heuristic allocates space to sibling nodes in a manner proportional to their importance in the text, as indicated by the discourse relations in the tree. Let's consider a node with 3 children: one nucleus, and two satellites, a preparation and an elaboration. The nucleus is allocated half the available space, and the two satellites a quarter each, assuming preparation and elaboration have been declared with equal importance. At each level of recursion, the available space is further divided. This means that after a certain recursion depth, the algorithm reverts to producing a list of links to each node's content (similar to a page of search results).

Another heuristic allocates most of the available space to the nucleus, and provides hypertext links to each of the satellites. This is the algorithm used to produce the small-screen output in Figure 1.

These two heuristics provide generic mechanisms for allocating content to space. It is also always possible to define operators very specific to a domain or a device.

As a final remark on the information assembly stage, as the tree is not extended during this stage, strictly speaking, there is thus no further planning involved. The operators used are not plan operators proper. However, for the sake of convenience, we have chosen to express these assembly operators using a very similar syntax to our plan operators, and, like plan operators, they are declarative rules used by the VDP engine. This allows us to reuse our existing planning engine and plan processing code to perform the information assembly stage.

4 Realisation

As noted, the annotations created during information assembly are interpreted during the final realisation tree walk. This makes the realisation process a relatively simple annotation interpretation process, which allows us to have a very generic realisation module.

As an example, when processing the children of a node, the realisation module simply needs to refer to the *realisationOrder* feature annotation in each node to know the correct order in which to process the children. Similarly, the presence of *addToIndex* annotations instruct the realisation module to add a reference to a particular node into the index, and hyperlink *anchorText* and *linkTarget* annotations specify how to create hyperlinks according to the decisions made during information assembly.

The realisation process results in the content actually being placed in specific page instances which are then displayed in specified frames of the chosen layout template. These pages are generated in a device-specific syntax, such as HTML or WML.

5 Implementation Status

This work is being carried out in the context of our work on contextualised information retrieval and delivery. The Myriad framework has been implemented and is being exploited in a number of domains and applications – e.g., Tiddler (Wilkinson *et al.*, 2000; Paris *et al.*, 2001), PERCY (Paris *et al.*, 2003), DFDMSA (Colineau and Paris, 2003, Colineau *et al.,* 2004a) and skil (Müller-Tomfelde *et al.*, 2004). The framework continues to be enhanced and extended.

The work presented here has recently been added into the framework and is currently being tested in an application requiring the automatic production of tailored reports.

Parts of the framework and the approach have been evaluated (e.g., Paris *et al.*, 2001; Paris *et al.*, 2003, Wilkinson and Wu, 2004; Wu *et al.*, 2004). We intend to perform further evaluations.

6 Conclusion

In our work, we are concerned with delivering information consistently and coherently across heterogeneous devices. To do so, we propose an approach which allows a presentation to be planned once, and yet be delivered appropriately on the delivery medium of choice.

In this paper, we distinguished between two types of decisions that must be made during the presentation planning stage of an information delivery system: *local* decisions, which are based only on the content or features of the *node* itself, and *global* decisions, which are based on the *entire structure* of the discourse tree.

We presented a generic algorithm for making global decisions, driven by the discourse tree structure and contextual characteristics. This algorithm, implemented through declarative operators, complements the more traditional local presentation decisions made through presentation planning.

7 Acknowledgements

We wish to thank Keith Vander Linden for his ideas embodied in the first space allocation algorithm described as part of this work, and for his work on the Myriad framework. We also thank other members of the Information Engagement group at the CSIRO ICT Centre who have helped us implement the algorithm, in particular MingFang Wu and Akshay Bhurtun.

References

Chincholle, D. (2000). Designing effective mobile services on small communication devices. Tips and Techniques. Tutorial notes in *Proceedings. of OzCHI 2000: Interfacing Reality in the New Millennium*. December 4-8, 2000. Sydney.

Colineau, N. and Paris, C. (2003). Task-Driven Information Presentation. *In Proceedings of the 2003 Annual Conference of the Computer-Human Interaction (OZCHI'03)*, the Special Interest Group of the Human Factors and Ergonomics Society of Australia, Brisbane, Australia, Nov 25-28, 2003.

Colineau, N., Lampert, A. and Paris, C. (2004a). Task-Sensitive User Interfaces: grounding information provision within the context of the user's activity. In *Proceedings of the working conference on Advanced Visual Interfaces (AVI) Conference*, Gallipoli, Italy, May 25-28 2004, 218-225.

Colineau, N., Paris, C. and Wu, M. (2004b). Actionable Information Delivery. In *Revue d'Intelligence Artificielle* (RSTI – RIA), Special Issue on Tailored Information Delivery, 18(4), 549-576. September 2004.

Freed, N and Borenstein, N. (1996). Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types – Published as "Request For Comments: 2046".

Hovy, E. (1988). Planning coherent multisentential text. In *Proceedings. of the 26th Conference of the ACL*, Buffalo, NY, 163-169.

Hovy, E., and Arens, Y., (1991). Automatic generation of formatted text. In *Proceedings of the 8th Conference of the American Association for Artificial Intelligence.* 92 – 96. Anaheim, CA: AAAI.

Klyne,G., Reynolds, F., Woodrow, C., Ohto, H., Hjelm, J., Butler, M. and Tran, L. (2004). *Composite Capabilites/Preference Profiles (CC/PP): Structure and Vocabularies 1.0*. W3C Recommendation 15 January 2004.

Mann, W. and Thompson S. (1988). *Rhetorical Structure Theory: Towards a Functional Theory of Text Organization.* In *Text*, 8(3), 243-281, 1988.

McKeown, K.R. (1985). *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text.* Cambridge University Press, Cambridge, England.

Moore, J. and Paris, C. (1993). Planning text for advisory dialogues: Capturing intentional and rhetorical information. In Computational Linguistics, 19(4), 651-694.

Müller-Tomfelde, C, Paris, C. and Stevenson, D. (2004). Interactive Landmarks: Linking Virtual Environments with Knowledge-Based Systems. To appear in the *Proceedings of the 2004 Annual Conference of the Computer-Human Interaction (OZCHI'04)*, the Special Interest Group of the Human Factors and Ergonomics Society of Australia, Wollongong , Australia, November 22-24, 2004.

Paris, C. (2002). Information Delivery for Tourism. In *IEEE Intelligent Systems* Intelligent Systems for Tourism: Trends and Controversies, Stephen Staab and Hannes Werthner (eds), November/December, 2002. pp 53 – 66, 2002.

Paris, C., Wan, S., Wilkinson, R. and Wu, M. (2001). Generating Personal Travel Guides – and who wants them? In *Proceedings of the International Conference on User Modelling (UM2001);* M. Bauer, P. Gmytrasiewicz and J. Vassileva (eds). Sonthofen, Germany, July 13-18, 2001. 251-253.

Paris C, Wu M, Vander Linden K, Post M, and Lu S. (2004). Myriad: An Architecture for Contextualized Information Retrieval and Delivery (2004). In *AH2004: International Conference on Adaptive Hypermedia and Adaptive Web-based Systems*. August 23-26 2004, The Netherlands. pp.205-214.

Paris, C., Wu, M., Vercoustre, A-M., Wan, S., Wilkins, P. and Wilkinson, R. (2003). An Empirical Study of the Effect of Coherent and Tailored Document Delivery as an Interface to Organizational Websites. In *Proceedings of the Adaptive Hypermedia Workshop at the 2003 User Modelling Conference*, Pittsburgh, USA, June 22, 2003. 133-144.

Wilkinson, R., Lu, S., Paradis, F., Paris, C., Wan, S. and Wu, M. (2000). Generating Personal Travel Guides from Discourse Plans. In Proceedings of International Conference on Adaptive Hypermedia and Adaptive Web-based Systems. Trento, Italy, August 2000.

Wilkinson, R and Wu, M. (2004). Evaluation Experiments and Experience from the Perspective of Interactive Information Retrieval, In the *Proceedings of the Third Workshop on Empirical Evaluation of Adaptive Systems*, in conjunction with AH2004. August 23-26, The Netherlands. pp.216-225.

Wu, M., Muresan, G., McLean, A. Tang, M., Wilkinson, R. ,Li, Y., Lee, H. and Belkin, N. (2004). Human versus Machine in the Topic Distillation Task, In the *Proceedings of the 27th SIGIR Conference on Research and Development in Information Retrieval*, July 25th-29th 2004, Sheffield, UK. pp. 385-392.