Towards Interaction Ontologies for Mobile Devices Accessing the Semantic Web

Pattern Languages for Open Domain Information Providing Multimodal Dialogue Systems

Daniel Sonntag
DFKI - German Research Center for Artificial Intelligence
Stuhlsatzenhausweg 3
Saarbrücken Germany
sonntag@dfki.de

ABSTRACT

We try to develop a communication basis between interface designers, natural language processing experts and scenario experts for multimodal dialogue systems. The question we address is, how to represent different kinds of results (information design) and interaction metaphors (interaction and presentation design) which are created and presented to the user during the dialogue. We present a pattern language for interaction design and show its strength to achieve a common data model. The common data model helps to handle the great complexity in the user-system interaction of our multimodal dialogue system SmartWeb, which provides Question Answering (QA) functionality and access to the Semantic Web. We motivate the construction of Interaction Ontologies as a natural extension for Semantic Web-based applications.

Keywords

Multimodal HCI, Dialogue Systems, Question Answering, Interaction Design, Presentation Design

1. INTRODUCTION

Dialogue systems are very complex human-machine interfaces where a lot of interaction is to be expected. Even simple question-answer patterns are often studied in terms of dialogue grammars [16] or dialogue games [9]. Dialogue systems normally encompass a variety of processing components, which need to handle a complex communication structure. Many integration frameworks have been developed to handle these complex system types [23, 3, 8, 21]. Communication is even more complex in a multimodal setting, where apart from normal text input and output, other input modalities such as speech do count as input.

Information seeking multimodal dialogue systems can be used to build up an extension to a QA system[1] in an interactive setting. Accordingly, there is a strong relationship between two major dialogue acts, the user question or query, and the presentation of system results. All other dialogue moves, e.g. user corrections, and processing steps are somewhere in between. Almost all dialogue moves initiated by the system require a presentation on the interaction device, or an input from the user. The difficulty is, apart from deciding on the actual information content to be presented, what kind of interaction metaphors and sequences should be created and presented to the user, giving interface designers, software architecture designers, information retrieval experts, scenario experts and natural language processing (NLP) experts a common language of necessary dialogue descriptions, message communication structures, and finally a language of common design patterns to be implemented in the course of development.

With the help of user studies [17], preferred metaphors for interaction and interaction sequences can be selected for realisation. This methodology turned out to be the basis for communication between interface designers and scenario experts in related large-scale dialogue system projects [20]. In the SmartWeb system, which is heavily influenced by recent Semantic Web developments [6], we push the idea of dialogue descriptions a bit further, to provide them with a formal representation.

In short, SmartWeb [26, 19] aims to develop a context-aware, mobile and multimodal interface to the Semantic Web. In the main scenario, the user carries a smartphone and is able to pose multimodal open-domain questions using speech, pen and gesture, among other input modalities. The user input is transmitted via UMTS or WLAN to a back-end server, where the multimodal recogniser, the Semantic Web access sub-systems and the dialogue manager for result

[1] Question Answering is the task of finding answers to natural language questions by searching large document collections. Unlike information retrieval systems, question answering systems do not retrieve documents, but instead provide short, relevant answers located in small fragments of text.

generation resist.

In this connection, we designed a pattern language for interaction design which mainly focuses on providing a common terminology for all interaction- and especially presentation elements presented to the user. All patterns that we collect for interaction and presentation issues can be structured into an *Interaction Pattern Language*.

In section 2 we explain the concepts behind Pattern Languages and show our implementation dedicated to multimodal result presentation. These realised elements of the Pattern Language are selected by the different information providing sub-modules, depending on the context information of the dialogue. These dialogue system modules decide either on the visual elements to be presented (headings, labels, colours, fonts, layouts, icons, etc.) or the information content of specific elements. Section 3 describes related work. In section 4 we conclude the presented work and outline, how we plan to expand the pattern language for subsequent integrations of the whole dialogue system in the Semantic Web context.

2. A PATTERN LANGUAGE FOR MOBILE DEVICE INTERACTION DESIGN

The original concept of pattern languages was invented by the urban architect Christopher Alexander in the 70's. The patterns he described define promenades, gardens, public places, and the design of single buildings and their alignment. Years later, patterns in HCI (Human Computer Interfaces) [1] and object-oriented application frameworks [11, 5] have emerged. Recent topics are patterns for interaction design. We follow the formal syntactic definition of a pattern language given in [4].

A pattern language is a directed acyclic graph $PL = (\mathbf{P}, \mathbf{E})$ with pattern nodes P_n and edges E_n. A pattern P *references* a pattern Q, if there exists an edge E, with $E = (P, Q) \in \mathbf{E}$. P is then the *context* of Q. This relationship defines a hierarchy on the patterns, although the relation itself is very fuzzy. Each pattern P is meant to consist of a set of properties such as its name, given solutions and examples.

User Interface Design Patterns such as trees, double lists, fisheye views, and the Global Undo Operation for example, are well known to everybody dealing with text editors or other graphical user interfaces. Especially in advanced graphical computer interfaces, data navigation, data selection and data input are combined with interaction metaphors. The fisheye view design pattern is also a good example for an interaction metaphor. Another example is a pattern $P_{progressbar}$ describing the problem whether or not a system operation is still being performed and how much longer the user will need to wait.[2] The solution in $P_progressbar$ is to indicate the user that the application is still working and to give an indication of the progress. The context of $P_{progressbar}$ is that the user wants to know if an operation is still being performed as well as how much longer the user will need to wait. In our *Interaction Pattern Language*, $P_{progressbar}$ would be referenced by the top-level pattern *Provide status information to the user.*

[2]See http://www.welie.com/patterns/gui/progress.html for the fully specified pattern by Martijn van Welie. It is to be noted that in this language *problem* refers to what we call *context*.

2.1 Interaction Pattern Language in SmartWeb

We describe how we constructed a pattern language for mobile interaction design, whereby the theoretical and practical benefits of this transformation in the context of our Semantic Web connected dialogue system SmartWeb is in focus.

In our multimodal dialogue system SmartWeb, the user interface is a smartphone (PDA). The display size of this mobile device is small (320*240 pixel). The pocket computer has very limited computational power. Nevertheless the user should be able to interact with the system in different modalities such as speech and gesture and refer to the displayed results for further inspection or posing a new query. Interface consistency will be provided by the interplay of navigation, layout design and the interaction metaphors. We believe that many communicational aspects are dependent on presentational aspects. Therefore interaction in multimodal system ties aspects of communication and the synchronisation of input/output messages close together. All interaction channels (most prominent, the output channels) must be synchronised to render the multimodal information to be presented. To meet the challenge, we need a model of interaction, especially presentation elements and the content they present. Here, we focus on the presentation elements: In our dialogue scenario, many results are naturally presented auditory and graphically (synchronously). We plan to exploit the graphical display as much as possible. Accordingly, all results will be presented at least on screen. We select an appropriate system response according to the type and focus of the user statement. This helps building human-computer interactions in a manner which is natural for humans. Unfortunately, it assumes, as mentioned before, a conceptual theory about the semantics of the utterance, some sort of understanding (natural language understanding) depending on the context (discourse memory), and leads finally to the question, what kind of responses are appropriate to be presented. The last point concerns both the aspects of content selection, medium selection and the visual presentation metaphors engaged, for which we construct a pattern language.

The pattern language we investigate is dedicated to presentational issues, presenting system status, intermediate results and final results. Since the user interacts via the smartphone, the pattern language should also reflect mobile interface heuristics (Every pixel counts, little navigation, restrict number of round trips, etc.). We identified six top-level patterns for interaction design to be realised:

1. Apply basic mobile user interface principles (common Look and Feel).
2. Allow for feedback from user input.
3. Offer correction possibilities.
4. Ensure interface simplicity by progressive disclosure.
5. Provide status information to the user.

6. Declare and select presentation layout patterns according to information content and dialogue act.

In all interactions and presentation situations, appropriate layout patterns have to be selected from a corresponding knowledge base and filled with dynamic content. Considering as example extract (see figure 1) we expose the benefits of the pattern language we have build up. Figure 1 outlines the extract of an interaction pattern language. For illustration, we restrict our patterns to provide basically a name to refer to its central idea.

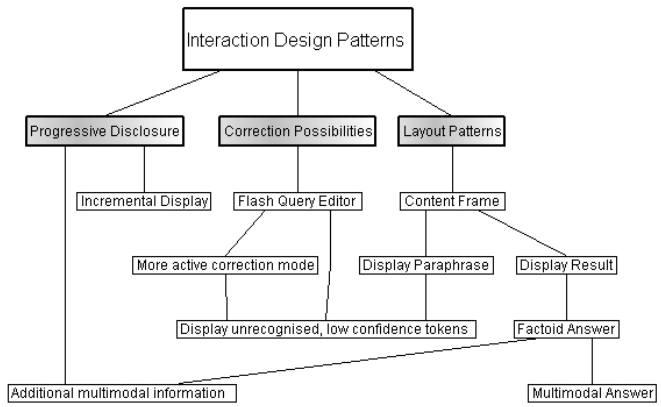

Figure 1: SMARTWEB **Interaction Design Patterns**

We briefly discuss the *Display Result* pattern in the following: It's context is the *Content Frame* of our GUI. To understand this, we show a complete realised layout in figure 2. The user can navigate through additional multimedia results. On the interaction device, several interaction patterns are realised concurrently, mainly on the screen. The *Display Result* pattern is realised in the area between the dotted lines. According to our patterns language, the *Display Result* pattern is referenced by the *Factoid Answer Pattern*. This pattern is thus realised as a special visual element within the content frame (2). The *Factoid Answer* Pattern is referenced by the *Additional Multimodal Information* pattern, its realisation is a media screen, where pictures and videos can be shown. In addition, the *Additional Multimodal Information* pattern has got the *Progressive Disclosure* pattern as context. This indicates that not all information elements presented by the realisation of the *Additional Multimodal Information* pattern are displayed concurrently. The media screen only displays one medium at a time, and with the help of the on-screen or on-device buttons (1) and (3), the user can navigate through different media instances which count as additional multimodal information to the factoid answer, such as pictures and supportive newspaper text paragraphs.

With the use of our pattern language we provided our designer parties with the necessary vocabulary to implement layout and software fragments for presenting these patterns. The pattern language (figure 1) turned out to be a good tool to lead the designers from large-scale design patterns to specific patterns about details. More importantly, the formal

Figure 2: Graphical layout for multimodal question answering functionality. Question: *When did Germany win the [football] world cup for the first time?* **Factoid Answer:** *1954*

representation we presented offers a common terminology of the different interaction patterns to be rendered on the mobile device in different presentation situations. It is important to note, that all graphical interface elements realised as pattern language instances have their dedicated position on the device. This avoids a planning mechanism for screen positions of visual elements and allows for predictable display results. The imposed constraints for overlapping elements and context dependency are part of the application logic of a presentation module. Within our dialogue system, the module REAPR (REaction And PResentation) coordinates which information is to be presented in which form, at each dialogue stage. The pattern language allowed us to create very specific and adaptable multimodal presentations. We created a simple Map function to assign each dialogue state (we are able to detect) to the appropriate graphical realisation in form of realised patterns of our pattern language.

For example, the dialogue state *Multimodal Result Presentation* (as shown in figure 2) is implemented by

```
Multimodal_Result_Presentation ->
{Flash_Query_Editor, Factoid_Answer,
Additional_Multimodal_Information}
```

As a second example, the dialogue state *User Correction* is implemented by

```
User_Correction -> {More_active_correction_mode,
Display_unrecognised_tokens}.
```

In the Semantic Web context, the correction of flawed speech recognition output is of paramount importance. The predominant input modality of our handheld scenario is speech. On the other hand, the queries for a Semantic Web search have to be as accurate as possible. Practically this means a lot of manual corrections to be done by the user. Therefore we display the best recognition result and offer the user a big editor window with advanced handwriting correction and menu selection possibilities. Confer the upper part of figure 3 to get an impression.

As a third example, in order to show the user, which semantic query is being send to the Semantic Web knowledge servers, the Semantic Query can be paraphrased in simple *attribute:value* form. For this purpose, the dialogue state *User Query* is mapped onto

```
User_Query ->  {Display_Paraphrase}.
```

Confer the lower part of figure 3. We experimented with different representations for the *attribute:value* pairs specifying the Semantic Web query more explicitly. In most cases, default values have to be introduced. In the solution presented, the paraphrase is of rather natural language form, entailing the concepts *Mannschaft*(team), *Land*(country), *Fussball*(football), and *Maenner*(men) out of our football domain ontology which entails the fact base in highly structured form. We choose (at least for demonstration) not to present the paraphrase in strict *attribute:value* form, nor deeply structured RDF form to the user, since the outstanding characteristic of the interface is its use of natural language. The generation engine we use for generating textual results (cf. similar approaches in [2]) could also be used for generating paraphrases. In addition, we highlighted the most important Semantic Web query terms, that the user is able to see the important terms at first glance, without the need for restricting the natural language syntax.

2.2 Implementation on the PDA

In our SmartWeb dialogue application, the mobile device renders the presentation in a Macromedia Flash Projector[3], which ships with a separate presentation authoring environment in which we design all visual elements. Therefore the usage of a pattern language as described seems most appropriate to query its structure for interaction elements, which correspond to actual Flash- or other software pieces. These pieces only need a few additional content information, such as text in a text field, to render a complete multimodal result on the smartphone screen.

In our case, the interaction elements have been implemented in Flash Actionscript 2.0 [14]. Since the resulting Flash-movies are portable to desktop computers and internet browsers, we are able to run and test our application in the same manner on the mobile end device and a normal computer. Hence interface consistency is guaranteed over multiple devices, last but not least because of our general multimodal dialogue setting for user input - the user can easily switch to the keyboard of his home computer to write down questions. Hence, interface portability and consistency is both

[3] www.macromedia.com

Figure 3: Graphical layout implementation for user queries to be answered by Semantic Web knowledge servers. The Interaction Design Patterns *More active correction Mode* **and** *Display Paraphrase* **are displayed concurrently. The question** *Who was world champion in 1990* **results in the augmented paraphrase** *Search for: World champion team or country in the year 1990 in the sport of football, division men.*

provided by system independent presentation implementation (as Flash movies) and the multimodal input setting, where users can use the input modality which is best in the user situation (confer using your handheld in the office vs. using while driving a car), and input device restrictions or opportunities: The computer has no sound-card installed for speech recognition, or the laptop's keyboard is a lot easier to use than the PDA's keyboard.

3. RELATED WORK AND DISCUSSION

We discuss the use of semantics to generate natural (dialogue) human-machine interactions and presentations. In a very special application domain, in which the dialogue interface is a smartphone, restriction yields in a configuration, were related work is very rare (By the way, the configuration also implies a lot of constraints on suitable dialogue components.). A practitioners guide on designing software for the mobile context [12] is equally important than recent work toward ontology-driven discourse [7]. Nevertheless the latter is of great importance to visionary work. In [7], a presentation generation system with Semantic Web technology is in focus and emphasises the importance of (1) to be effective in conveying relevant domain semantics and (2) to model generally applicable presentations. Unfortunately, in the context of mobile devices and information providing dialogue systems for answering questions, a generally applicable approach is not appropriate, since the interaction, as exemplified, is of a very special sort. We therefore think that ad hoc generation of graphical elements and layouts should be avoided. Automated multimedia presentation generation is more appropriate on larger screens where much more visual elements can be presented and the accuracy of manual authoring is less important. This is why a planning approach for presentation design as used in Smartkom [18, 25] can be labelled as unsuitable. SMIL 2.0 (Synchronized Multimedia Integration Language) [24] is an XML-based language that allows authors to write interactive multimedia presentations, and is very valuable in planning environments where the complete content displayed on screen is dynamically build up. For our actual system we think that SMIL syntax and semantics markup presents unnecessary modelling effort between the server-side presentation module and the rendering module on the device. A related question is, how to scale down a generation approach in case of erratic network connection to the dialogue and presentation server. Since this is a general problem in distributed presentation system, our pattern language approach suffers from the fact, that at least the information content must be provided by the server for presentation. On the other hand, the implemented interaction patterns are designed to allow for interaction even if a network connection breaks away. For example, the query editor can be used offline with the PDA keyboard, although the server-based speech recognition system is not available.

Generating Hypermedia Presentations (e.g. [22]) with explicit document structure is the right context of multimodal discourse history to present the course of the dialogue as a whole. The idea here is to model single dialogue interactions and intermediate presentations in the context of dialogue history. These discourse models are often expressed in terms of RST (Rhetorical Discourse Structure) [13]. However, multimodal media items are more difficult to relate to each other, even more difficult in a setting where the exact relationship to each other is not known. Even if all media objects are presented as Semantic Web instances stemming from well-described data sources (e.g. logic-based techniques in data integration [10]), effective multimodal presentation remains a challenge on a meta level. Last but not least, because semantic-based data integration frameworks for multimedia data are still missing, templates and patterns with a relatively fixed multimodal fusion/fission and presentation structure seem to be the adequate means. Our pattern language as a consequence of a language description as such requires the designers to agree on the language items and develop the language in an early project phase, which might result in a considerable additional effort not acceptable for small-scale projects.

4. CONCLUSION AND OUTLOOK

We have presented a pattern language for interaction design in a multimodal dialogue system. The realisation on the mobile end device shows the flexibility of this approach, even though we do not generate layouts and graphical elements ad hoc.

The idea of a pattern language as starting point seems to come out even. User interface principles include graphic design and visual aesthetics in the theory of graphical representation. For large-scale projects in the mobile context we recommend to explicitly state and build all interaction and presentation elements in a pattern language to be realised as software snippets.

In order to take full advantage of an ontological description to improve interaction patterns, we plan to extend the pattern language we describe into an interaction ontology, in order to bring in architectural properties of SmartWeb as a Semantic Web application. Ontologies are used to provide a formal semantic interpretation of component information to be integrated into a common data model. With the help of an interaction ontology, we incorporate visualisation templates and user interaction elements into this data model, too. Therefore the overall benefit is more than delivering a formal representation to guide implementation issues, for example. Integrated into a common data model, the interaction ontology enriches content-based fusion, selection and suitable presentation of system reactions enabled by a common semantic (ontological) framework. As a result, an ontology of interaction patterns fully integrates into this approach. In an interaction ontology, properties of patterns can be expressed in a much more elegant way by dint of ontological descriptions. The interesting part is the possibility to transform pattern languages like the one illustrated in figure 1 into an ontology, adding some structural commitments in order to model concepts and relations pertinent in the domain in a richer more declarative language. This makes the domain assumptions more explicit and operational. Since an ontology focuses on the structural properties of a concept, we plan to first transform the pattern language into a ontological taxonomic structure. Thereby the fuzzy *referencing*-relation from one pattern to the other pattern will receive a clear distinction between taxonomic sub-classes and aggregations, which bundle sets of possible interactions as attribute slots. Furthermore the possibility to differentiate between class properties and instance properties [15] will be explored in the context of interaction patterns and their specific software implementations.

5. ACKNOWLEDGEMENTS

I would like to thank our partners in SmartWeb and Norbert Reithinger for useful comments on earlier versions of this article. This research was funded by the German Federal Ministry for Education and Research under grant number 01IMD01A. The responsibility for this article lies with the author.

6. REFERENCES

[1] Apple. *Apple Computer. Macintosh Human Interface Guidelines.* Addison-Wesley, 1992.

[2] T. Becker and S. Busemann, editors. *Impacts in Natural Language Generation: NLG between Technology and Applications. Workshop at Schloss Dagstuhl, Germany*, number D-00-01 in DFKI Document, Saarbrcken, 2000. DFKI.

[3] K. Bontcheva, V. Tablan, D. Maynard, and H. Cunningham. Evolving GATE to Meet New Challenges in Language Engineering. *Natural Language Engineering*, 10, 2004. Special issue on Software Architecture for Language Engineering.

[4] J. O. Borchers. A pattern approach to interaction design. In *Proceedings of the conference on Designing interactive systems*, pages 369–378, New York City, 2000.

[5] J. W. Cooper. *The Design Patterns, Java Companion.* Addison-Wesley, 1998.

[6] D. Fensel, J. A. Hendler, H. Lieberman, and W. Wahlster, editors. *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press, 2003.

[7] J. Geurts, S. Bocconi, J. van Ossenbruggen, and L. Hardman. Towards ontology-driven discourse: From semantic graphs to multimedia presentations. In *ISWC 2003 Second International Semantic Web Conference*, Sanibel IslandFL, USA, 2003.

[8] G. Herzog, A. Ndiaye, S. Merten, H. Kirchmann, T. Becker, and P. Poller. Large-scale Software Integration for Spoken Language and Multimodal Dialog Systems. *Natural Language Engineering*, 10, 2004. Special issue on Software Architecture for Language Engineering.

[9] J. A. Levin and J. A. Moore. Dialogue-games: Metacommunication structures for natural language interactions. *Cognitive Science, 1, (4), 395-420*, 1977.

[10] A. Y. Levy. Logic-based techniques in data integration. In J. Minker, editor, *Workshop on Logic-Based Artificial Intelligence, Washington, DC, June 14–16, 1999*, College Park, Maryland, 1999. Computer Science Department, University of Maryland.

[11] T. Lewis, editor. *Object-oriented Application Frameworks*. Prentice-Hall, 1996.

[12] R. Longoria, editor. *Designing software for the Mobile Context. A Practitioner's Guide*. Springer, 2004.

[13] W. C. Mann and S. A. Thompson. Rhetorical structure theory: A theory of text organization. Technical report, ISI: Information Sciences Institute, Los Angeles, CA, ISI/RS-87-190, 1987.

[14] C. Moock. *Essential ActionScript 2.0*. O'Reilly, 1. edition, 2004.

[15] N. F. Noy and D. L. McGuinness. Ontology development 101: A guide to creating your first ontology. Technical report, Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics, 2001.

[16] L. Polanyi and R. Scha. A syntactic approach to discourse semantics. In *Proceedings of the 10th international conference on Computational linguistics*, pages 413–419, Morristown, NJ, USA, 1984. Association for Computational Linguistics.

[17] T. Portele, S. Gorony, M. Emele, A. Kellner, S. Torge, and J. te Vrugt. SmartKom-Home—An Advanced Multi-Modal Interface to Home Entertainment. In *Proc. of Eurospeech'03*, Geneva, Switzerland, 2003.

[18] N. Reithinger, J. Alexandersson, T. Becker, A. Blocher, R. Engel, M. Löeckelt, J. Müeller, N. Pfleger, P. Poller, M. Streit, and V. Tschernomas. Smartkom - adaptive and flexible multimodal access to multiple applications. In *Proceedings of ICMI 2003*, Vancouver, B.C., 2003.

[19] N. Reithinger, S. Bergweiler, R. Engel, G. Herzog, N. Pfleger, M. Romanelli, and D. Sonntag. A look under the hood - design and development of the first smartweb system demonstrator. In *To Appear: Proceedings of ICMI*, Trento, Italy, 2005.

[20] N. Reithinger and G. Herzog. An Exemplary Interaction with SmartKom. In Wahlster [25].

[21] N. Reithinger and D. Sonntag. An integration framework for a mobile multimodal dialogue system accessing the semantic web. In *Proc. of Interspeech'05*, Lisbon, Portugal, 2005.

[22] L. Rutledge, M. Alberink, R. Brussee, S. Pokraev, W. van Dieten, and M. Veenstra. Finding the story - broader applicability of semantics and discourse for hypermedia generation. In *Proceedings of the 14th ACM conference on Hypertext and Hypermedia, Nottingham, UK,*, 2003.

[23] S. Seneff, R. Lau, and J. Polifroni. Organization, Communication, and Control in the Galaxy-II Conversational System. In *Proc. of Eurospeech'99*, pages 1271–1274, Budapest, Hungary, 1999.

[24] W3C. Synchronized multimedia integration language (smil 2.0). http://www.w3.org/TR/2005/REC-SMIL2-20050107/, January 2005.

[25] W. Wahlster, editor. *SmartKom: Foundations of Multimodal Dialogue Systems*. Springer, Berlin, 2004.

[26] W. Wahlster. Smartweb: Mobile applications of the semantic web. In P. Dadam and M. Reichert, editors, *GI Jahrestagung 2004*, pages 26–27. Springer, 2004.