Discourse-based Interaction Models for Recommendation Processes

Dominik Ertl, Hermann Kaindl, Edin Arnautovic, Jürgen Falb, and Roman Popp
Vienna University of Technology,
Institute of Computer Technology
A–1040 Vienna, Austria
{*ertl, kaindl, arnautovic, falb, popp*}*@ict.tuwien.ac.at*

***Abstract*—Manually creating recommendation processes and their user interfaces usually requires a lot of effort. Therefore, we propose high-level interaction design and automated generation of user interfaces for building dialogue-based product advisors. For this purpose, this work introduces discourse models as interaction models used for modeling recommendation processes. Such discourse models refer to domain-of-discourse models, which represent, among other concepts, the products (and their related product categories) that are to be recommended. So, this paper presents how discourse-based interaction models and their corresponding domain-of-discourse models can be used for modeling recommendation processes.**

Keywords*—*Interaction design; discourse model; recommendation process.

I. INTRODUCTION

Dialogue-based product advisors have become very popular on the Web in the context of e-commerce. Web users try to inform themselves about certain products, and many companies use recommendation processes to offer information and stimulate demand for their products. The range of products to be recommended is nearly unlimited. There are several underlying principles to build such a dialogue-based product advisor [2]. The possible interactions between a user and the dialogue-based product advisor, however, are often programmed directly into a graphical user interface (GUI) without a high-level interaction design.

This paper introduces discourse models as interaction models of recommendation processes. We focus on processes in knowledge-based recommender systems that allow for *preference elicitation* and do not involve user modeling and/or profiling. We use discourses to model the possible interactions between a human and a computer-based recommendation process as dialogues between them.

Figure 1 shows an overview of our approach (concrete examples can be found below). It illustrates that such a discourse model refers to an underlying model of the domain-of-discourse, which represents the possible content of the dialogues (concrete values of answers). In principle, such domain-of-discourse models can be derived from (product) ontologies like *GoodRelations* [1]. Discourse models serve together with related domain-of-discourse-models as a basis for GUI generation (see, e.g., [6]). We also plan to generate

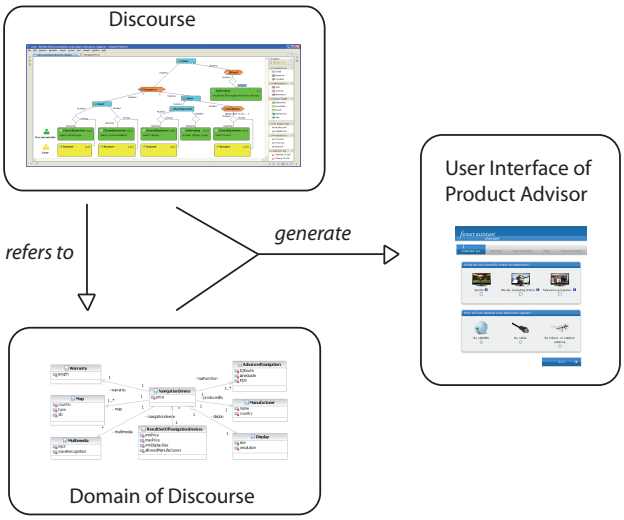

Figure 1. Overview of our approach.

GUIs for real recommendation processes to be used in commercial applications.

The remainder of this paper is organized in the following manner. First, we present background information about our discourse-based interaction models in general. Then we discuss related work. After that, we show how such a discourse model looks like for a recommendation process, using a class of recommendation dialogues for navigation devices as an example. Extending the same example, we finally show how a related domain-of-discourse model looks like.

II. BACKGROUND

Recommendation processes for preference elicitation have a structure of (partially ordered) sequences of questions and answers. They are potentially clustered into groups of questions/answers that belong together according to their semantics.

Discourse-based interaction models fit well for defining such structures. Our approach to discourse modeling is based on several theories of human communication from several fields [5]. The key ingredients of our discourse models are

 ISBN: 978-1-61208-117-5

Communicative Acts as derived from speech acts [9], *Adjacency Pairs* adopted from Conversation Analysis [7], and RST relations inherited from Rhetorical Structure Theory (RST) [8].

Communicative Acts (semi-structured messages with intention) represent basic units of language communication. Thus, any communication can be seen as enacting Communicative Acts: acts such as making statements, giving commands, asking questions and so on. Communicative Acts carry the intention of the interaction (e.g., asking a question or issuing a request).

Adjacency Pairs are sequences of talk-turns that are specific to human (oral) communication, e.g., a question should have a related answer. RST relations specify relationships among text portions and associated constraints and effects. The relationships in a text are organized in a tree structure, where the rhetorical relations are associated with non-leaf nodes, and text portions with leaf nodes. In our work, we use RST for linking Adjacency Pairs of Communicative Acts and further structures made up of RST relations. We have also included procedural constructs, to provide means to express a particular order during discourse execution, to specify repetitions or conditional execution of different discourse parts. While we only use a few types of Communicative Acts and RST relations yet, our modeling tool-kit has been sufficient for modeling even real-world discourses.

III. Related Work

Chen et al. [3] present interaction design guidelines for a critiquing-based recommender system that acts like an artificial salesperson. It engages users in a dialogue where users can provide feedback in the form of critiques to the sample items that were shown to them. The authors point out that the feedback, in turn, enables the system to refine its understanding of the user's preferences and prediction of what the user "truly wants". Our work targets on interaction design as well, however, we do not provide user prediction behavior of the system but models in terms of discourses.

Doyle and Cunningham [4] analyze the problem of deciding on the set of questions to ask in a session of navigation-by-asking recommender systems, including the issue of optimal ordering of the questions. Their work includes the evaluation of different question-selection criteria. Our modeling approach is more comprehensive, however.

IV. Discourse Models for Recommendation Processes

Figure 2 shows a discourse model for our running example, that models high-level interactions for recommendations regarding a product, in our case a navigation device. The diagram shows *Communicative Acts* of two participating actors, a human user who is the potential customer (light, yellow boxes), and the computer (dark, green boxes). More precisely, the user interacts with the dialogue-based Product Advisor according to this process.

The recommendation process is modeled as a sequence of several questions and related answers (*Adjacency Pairs*, shown as diamonds in Figure 2), which are supposed to help the user of the dialogue-based Product Advisor finding a navigation device that fits his or her wishes and needs. This is a process of preference elicitation. The overall procedural construct used in this model is a *Sequence*. More precisely, this construct does not directly link these *Adjacency Pairs* but trees of discourse relations that contain them.

In the first branch of the *Sequence* relation (left in the figure), a *Joint* relation combines two of these *Adjacency Pairs*. Joints below the Sequence cluster questions that hold a semantic relation. The first question gathers information on the price range, defining the minimum and maximum price that the user is potentially willing to pay. The second question elicits all manufacturers of navigation devices the user is interested in. This is a *closed* question since it provides all available choices of manufacturers.

In the second branch of the *Sequence* relation (right in the figure), another *Joint* relation combines two more *closed questions* about the voice recognition and mp3 functions. This second Joint clusters *multimedia* questions. The question about choosing voice recognition is additionally related with another *Adjacency Pair* that has an *Informing* as opening Communicative Act (here, no closing Communicative Act is needed). *Background* is an RST relation that optionally informs the human user on additional details about the subject matter, e.g., more information on voice recognition. Moreover, with the procedural construct *Condition* we define a condition that has to be fulfilled at runtime to have this branch considered. In the running example, the question about the mp3 function is only asked if the user has selected voice recognition. So, in this context of recommendation processes, such a *Condition* implements a rule for determining whether a certain question will be asked by the Product Advisor, i.e., whether it will be displayed in its GUI.

Of course, several more questions and answers would be included here in a real recommendation process. For the purpose of this running example, however, let us assume that this is the whole sequence.

The *Joint* at the top of the sequence has a procedural construct *IfUntil* with an *Informing* about all products that match the preference elicitations, if any. Whenever an answer is selected in any question of this recommendation process, the updated list of matching products (the results) is presented to the user. However, the discourse model does not contain the rules for matching of products at runtime. They have to be contained in the application logic of the dialogue-based Product Advisor. Still, it should be clear that subsequently selecting a product to buy, and paying it can also be modeled according to our approach.

 ISBN: 978-1-61208-117-5

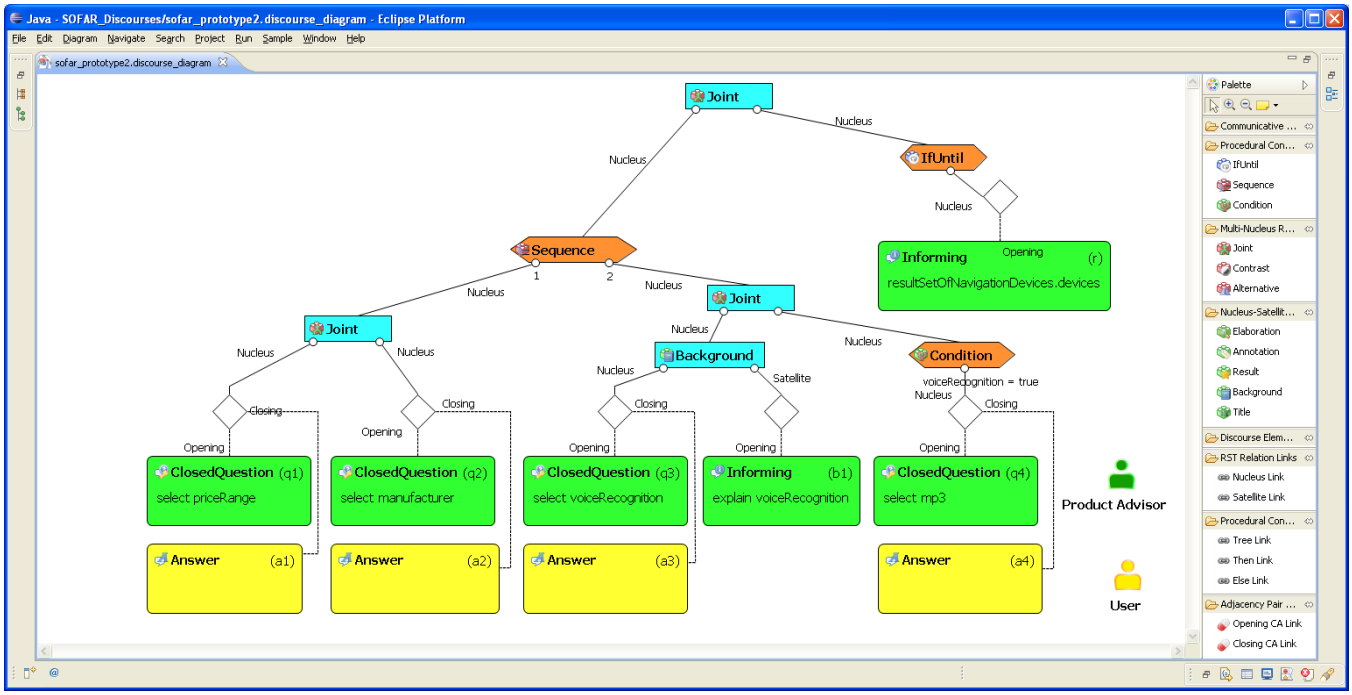

Figure 2. Discourse model representing a recommendation process for a navigation device.

In addition, *Communicative Acts* specify their propositional content, that refers to concepts in the domain of discourse. In Figure 2, these references are given in a short-hand notation, e.g., "select priceRange". These references link to model elements in the domain-of-discourse model in our approach.

V. Domain-of-Discourse Models for Recommendation Processes

A domain-of-discourse model may have been informed from a related product ontology. However, for the dialogues in the course of the recommendation process, not all concepts from the ontology are relevant. So, the domain-of-discourse model may be seen as the subset of an ontology that is sufficient to specify the content of the Communicative Acts within the defined discourses.

Figure 3 shows a UML class diagram for an example of such a domain-of-discourse model from the domain of navigation devices. Such types are defined more generally, e.g., in a product ontology. A NavigationDevice in our example model has the attribute *price*, parts like a *Display*, as well as references to other properties and functions, e.g., *Multimedia* functions. These other properties and functions can have their own attributes.

Now let us explain more technically what the references from the discourse model to this model mean, through expanding the short-hands from Figure 2:

- select priceRange: *min and max price from Range(Min(all NavigationDevice.price), Max(all NavigationDevice.price))*. The Product Advisor requests the user to set the *minPrice* and *maxPrice* attributes of the instance named *resultSetOfNavigationDevices* of the class *ResultSetOfNavigationDevices*.
- select manufacturer: *select many manufacturer from all Manufacturer*. The dialogue-based Product Advisor presents a set of all available Manufacturers to the user, who selects one or more of them to specify the list of manufacturers of interest. The Product Advisor sets the attribute *manufacturersOfInterest* of *resultSetOfNavigationDevices* with this list.
- select voiceRecognition: *select voiceRecognition from MultiMedia*. The Product Advisor presents a check box that the user can select to get a navigation device with or without voice recognition. The Product Advisor sets the attribute *voiceRecognition* to true or false.
- explain voiceRecognition: The Product Advisor presents a background text explaining the feature to support the user in her decision if she wants to get a navigation device with or without voice recognition.
- select mp3: *select mp3 from MultiMedia*. The Product Advisor presents a check box to the user. Thus, the user can select to get a navigation device with or without mp3 capability. The Product Advisor then sets the attribute *mp3* to true or false.
- resultSetOfNavigationDevices.devices: *display ResultSetOfNavigationDevices::resultSetOfNavigation Devices.devices*. The GUI displays all Devices which are referenced by the resultSetOf

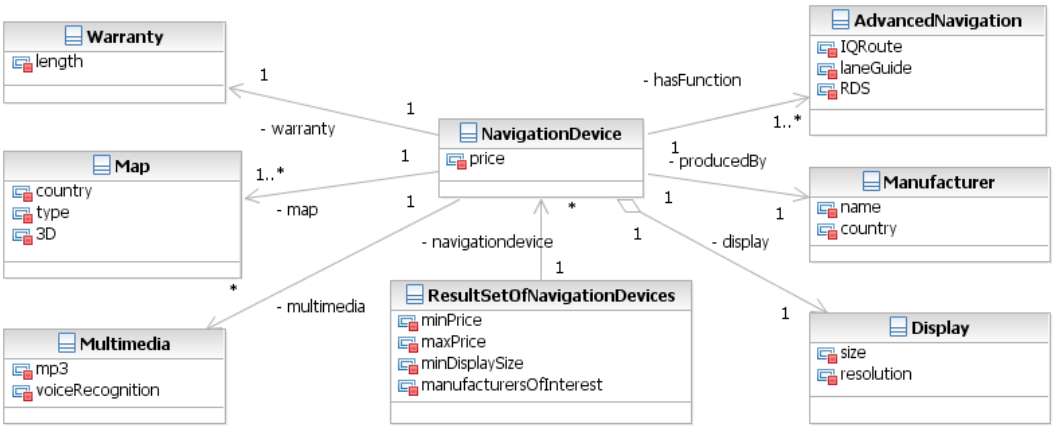

Figure 3. Domain-of-discourse model for a navigation device recommendation process.

NavigationDevices. This list contains only the devices fulfilling the defined constraints.

VI. Conclusion

In this paper, we show that and how a discourse model can represent a high-level interaction design of a recommendation process. Such discourse models are classes of dialogues that are possible between a human user and a Product Advisor that implements this recommendation process. Such a discourse model refers to a domain-of-discourse model in the sense that the latter specifies the content of the dialogues of the former.

A domain-of-discourse model may be part of a related ontology that is directly relevant for the dialogues. We currently work on support for extracting and deriving domain-of-discourse models from ontologies.

In our previous work, we have already generated multimodal user interfaces automatically from discourse models together with related domain-of-discourse models. We are currently working on feeding automatically generated structural GUI models into a real dialogue-based Product Advisor, that will generate its usual Web-based GUIs from them.

Acknowledgment

This research has been carried out in the SOFAR project (No. 825061), funded by the Austrian FIT-IT Program of the FFG and Smart Information Systems GmbH.

References

[1] Goodrelations: Last visited on December 10, 2010. http://www.heppnetz.de/projects/goodrelations/.

[2] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[3] L. Chen and P. Pu. Interaction design guidelines on critiquing-based recommender systems. *User Modeling and User-Adapted Interaction*, 19(3):167–206, 2009.

[4] M. Doyle and P. Cunningham. A dynamic approach to reducing dialog in on-line decision guides. In *Proceedings of the 5th European Workshop on Advances in Case-Based Reasoning (EWCBR '00)*, pages 49–60. Springer-Verlag: London, UK, 2000.

[5] J. Falb, H. Kaindl, H. Horacek, C. Bogdan, R. Popp, and E. Arnautovic. A discourse model for interaction design based on theories of human communication. In *Extended Abstracts on Human Factors in Computing Systems (CHI '06)*, pages 754–759. ACM Press: New York, NY, 2006.

[6] J. Falb, S. Kavaldjian, R. Popp, D. Raneburger, E. Arnautovic, and H. Kaindl. Fully automatic user interface generation from discourse models. In *Proceedings of the 13th International Conference on Intelligent User Interfaces (IUI '09)*, pages 475–476. ACM Press: New York, NY, 2009.

[7] P. Luff, D. Frohlich, and N. Gilbert. *Computers and Conversation*. Academic Press, London, UK, January 1990.

[8] W. C. Mann and S. Thompson. Rhetorical Structure Theory: Toward a functional theory of text organization. *Text*, 8(3):243–281, 1988.

[9] J. R. Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, Cambridge, England, 1969.

 ISBN: 978-1-61208-117-5