Emergent Conversational Recommendations: A Dialogue Behavior Approach*

Pontus Wärnestål, Lars Degerstedt, Arne Jönsson
Department of Computer Science
Linköping University, Sweden
{ponjo,larde,arnjo}@ida.liu.se

Abstract

This paper presents and evaluates a behavior-based approach to dialogue management, where a system's complete dialogue strategy is viewed as the result of running several dialogue behaviors in parallel leading to an emergent coherent and flexible dialogue behavior. The conducted overhearer evaluation of the behavior-based conversational recommender system CORESONG indicates that the approach can give rise to informative and coherent dialogue; and that a complete dialogue strategy can be modeled as an emergent phenomenon in terms of lower-level autonomous behaviors for the studied class of recommendation dialogue interaction.

1 Introduction

The purpose of a *recommender system* is to produce personalized recommendations of potentially useful items from a large space of possible options that is hard to manually browse or search. *Conversational* Recommender Systems (CRSs) approach user preference acquisition from a dialogue point of view, where preferences are captured and put to use in the course of on-going natural language dialogue. The approach is motivated by its aim to make interaction efficient and natural (Burke et al., 1997; Thompson et al., 2004), to acquire preferences from the user in a context when she is motivated to give them (Carenini et al., 2003), as well as to facilitate exploration of the domain and the development of the user's preferences (Wärnestål, 2005). A CRS's *dialogue strategy* to achieve these aspects of the interaction is thus crucial for its performance and usability. In particular, we are interested in exploring robust and emergent factual and preferential dialogue with recommendation capabilities.

This paper presents our behavior-based approach to dialogue management and reports on an evaluation of the CRS CORESONG's dialogue behaviors.

2 Dialogue Behaviors in Recommendation Dialogues

By a *dialogue behavior* of a dialogue agent, we understand a conceptual and computational functionality in the agent's dialogue strategy. Computationally, a dialogue behavior is coded into a *Dialogue Behavior Diagram* (DBD), that describes a state automaton where each state contains (one or more) commands and transitions with optional conditions. The DBD automaton is similar to the UML activity diagram.

DBDs invoke, and use, results from other software modules, denoted jointly as *external resources* (e.g. databases and recommender engines).

Four DBDs constitute the complete recommendation dialogue model: **Conventional**, **Direct Delivery**, **Indirect Delivery**, and **Interview**. A more detailed account of each of these behaviors are found in (Wärnestål et al., 2007).

Delivery Behaviors On a fundamental level, the goal for CORESONG (or any recommender system)

*This work is supported by the Swedish National Graduate School for Language Technology (GSLT), and Santa Anna IT Research.

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 63–66,
Antwerp, September 2007.

is to provide the user with a delivery, such as an explicitly requested piece of information from a database resource, or a recommendation from a recommender engine. The **direct** delivery typically uses a database that the user can query. In cases where a successful database call has been made (that is, a non-empty result set not larger than a predefined size limit is returned as response to a user's request), a delivery is written to the out stream. In order to support domain exploration and to drive the dialogue forward (Wärnestål, 2005), positive preference statements trigger the system to fetch information *related* to the preference from an external database resource (e.g. utterances S3a, S4a and S5a in Figure 1).

While the direct delivery behavior is functional for information-providing dialogue, and can cater for basic recommendations, it has been found that human-like recommendations occur in a slightly different fashion (Wärnestål et al., 2007). Therefore, the **indirect** delivery behavior has been designed (Wärnestål et al., 2007). It provides a *motivation* before presenting the actual recommendation (as exemplified by utterances S5b/S5c in Figure 1). Since the system cannot know whether the suggestion is previously familiar to the user, it delivers the recommendations in the form of questions (S5c).

Interview Behavior The purpose of the interview behavior is to collect relevant information about domain entity types (e.g. genres, artists or albums in the music domain) or items. This is a useful dialogue behavior in cases where deliveries cannot be completed due to e.g. ambiguous or incomplete user requests. It is also useful in preferential interviews used for recommendations, where the system acquires user preferences to be used by a recommender engine resource. A particular system may thus employ several interview DBD instances, connected to different external resources.

2.1 Emergent Dialogue

We view the dialogue system's complete behavior as emerging from the different DBDs. By *emergent* functionality in a system, we understand components that operate simultaneously in order to achieve a desired behavior. This is contrasted to hierarchical systems, where sub-functions are invoked from a central component or representation.

U1	What genres are there?
S2a	I found these genres in the database: [...].
S2b	I'd like to know more about your preferences.
S2c	What else do you like?
U2	I like the genre Rock&Roll
S3a	These artists belong to the genre Rock&Roll: [...].
S3b	I'd like to know more about your preferences.
S3c	What else do you like?
U3	What songs were made by Elvis Presley?
S4a	I found these songs by the artist Elvis Presley in the database: [...].
S4b	I'd like to know more about your preferences.
S4c	What else do you like?
U4	I like the album Live at Madison Square Garden
S5a	These songs belong to the album Live at Madison Square Garden: [...].
S5b	You might like the song Suspicious Minds because it is a Rock&Roll song by Elvis Presley.
S5c	Have you heard it?
U5	Yes
S6a	Ok. What do you think about it?
U6	It's good

Figure 1: *Dialogue log from* CORESONG *interaction (*RECOMMENDER *experiment configuration). [...] denotes lists of genres, artists, albums or songs. S = system, U = user.*

Our approach to dialogue system design is inspired by the layered subsumption architecture (Brooks, 1991) where layers correspond to behaviors that are organized hierarchically, and where higher-level behaviors can *subsume* lower-level layers by inhibition or modification.

A dialogue agent's complete strategy is described by a set of DBD instances that run as a DBD *strata machine*. The DBD strata machine streams input and merges each behavior's output (see Figure 2). There is no central representation of the complete dialogue, and the individual behaviors do not model each other since each DBD processes the incoming token stream autonomously. Therefore, the outputs from the DBDs need to be integrated (and typically reduced) into a coherent system turn, and is managed by two constructs in the Output Weaver: *behavior priority* and an *order heuristic*.

Behavior Priority DBDs are indexed with a priority and order the out statements accordingly (ascending order). The *request* with highest priority will be chosen. This hinders the occurrence of two requests back to the user which obviously could be confusing. The order of CORESONG's DBDs are (lowest to highest priority): Conventional, Direct Delivery,

Indirect Delivery, and Interview (Figure 2). DBD instances connected to the recommender engine have higher priority than those of the music database[1].

Order Heuristic Due to the behavior priority, there is only one request action available each turn. The order heuristic places this request at the end of the output, so that *informing* system action statements are guaranteed to precede the request. This guarantees that the constrain request (S2c) in the first system utterance in Figure 1 always occur after the direct delivery (S2a) even though their statements origin from different DBD instances.

3 Experiment

To validate the behavior based approach to dialogue management we conducted an "overhearer" experiment (Whittaker and Walker, 2004) by using four different behavior configurations of the CORESONG system (see Table 1). The reason for using the overhearer model is to avoid natural language interpretation problems (since the coverage of grammar and lexicon is not our focus), and letting personal music preferences that may not be covered by our recommender engine and database affect the subjects' experience of dialogue interaction. The experiment was run with 30 subjects.

3.1 CoreSong

Configuration of dialogue behaviors and attached external resources is easily done in CORESONG by switching DBD instances on or off. The two external resources used by the DBD instances are (a) a music information database and (b) a content-based recommender engine (Burke, 2002).

A DBD instance implementation consists of defining LookUp calls, and the surface realization of the action statements in the DBDs.

The Input Streamer (IS) feeds the interpretations of user input to each of the DBD instances in the DBD strata machine. Each DBD instance processes the input and writes to an out stream using the command out. The Output Weaver module (OW) then weaves together each DBD instance's output as outlined in Section 2.1.

[1]Note that interview and delivery behaviors of the same external resource are naturally designed to be mutually exclusive.

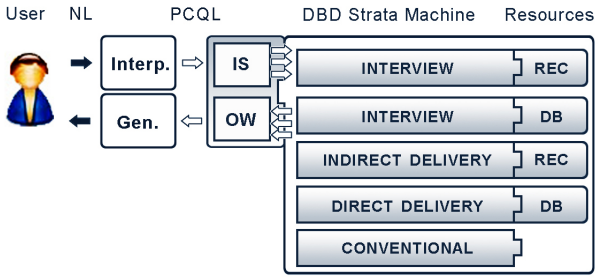

Figure 2: *The standard CoreSong behavior configuration, with database (DB) and recommender engine (REC), interview and delivery behaviors. Interp = Interpretation Module, Gen = Generation Module, IS = Input Streamer, OW = Output Weaver.*

Table 1: Experiment configurations. DD = Direct Delivery, IW = Interview, ID = Indirect Delivery, Db = Database, R = Recommender Engine.

Config.	DD(Db)	IW(Db)	DD(R)	ID(R)	IW(R)
Q-A	x	x			
BLUNT	x	x	x		x
PRYING				x	x
REC	x	x		x	x

Four different DBD instance configurations were used to generate the test dialogues, as shown in Table 1. The different configurations effectively modify CORESONG's complete dialogue strategy. Q-A, for example, with only the database resource, results in a question-answer system without recommendation capabilities, whereas the PRYING configuration supports a preference interview but with no power to deliver answers to factual requests. The BLUNT configuration has the power to deliver both database results and recommendations; but the recommendations are not delivered with motivations and follow-up questions as the indirect delivery (RECOMMENDER configuration) is designed to do. Figures 1 (RECOMMENDER) and 3 (BLUNT) exemplify the differences.

3.2 Procedure

Each subject was presented with the four test dialogues, one at a time, displayed in a web browser. For each of the dialogues they were asked to fill

U1	What genres are there?
S2a	I found these genres in the database: [...].
S2b	What else do you want to know?
U2	I like the genre Rock&Roll
S3a	These artists belong to the genre Rock&Roll: [...].
S3b	What else do you want to know?
U3	What songs were made by Elvis Presley?
S4a	These songs belong to the artist Elvis Presley: [...].
S4b	What else do you want to know?
U4	I like the album Live at Madison Square Garden
S5a	These songs belong to the album Live at Madison Square Garden: [...].
S5b	You might like the song Suspicious Minds.
S5c	What else do you like?

Figure 3: *Dialogue sample for the* BLUNT *configuration.*

out a questionnaire on a 5-point Likert-scale regarding their agreement with four statements, intended to determine *informativeness*, *preference modeling*, *coherence*, and *naturalness* of the dialogue excerpts. For example, the statement: "The system's utterances are easy to understand and provide relevant information" reflects informativeness (Whittaker and Walker, 2004).

4 Results and Discussion

In general, the participants considered the Q-A and RECOMMENDER configurations to have the highest informativeness (86.2% and 85.5% respectively). This is expected, since they both are equipped with the database direct delivery behavior. The PRYING configuration, lacking in database delivery functionality, received a lesser rating on informativeness. For our current work, the notion of coherence is of high importance, since this quality of the dialogue was thought to be at risk when abandoning a monolithic dialogue strategy model. It is interesting that the coherence measure is high for all configurations: PRYING (70.3%), BLUNT (79.3%), RECOMMENDER (84.1%) and Q-A (86.2%). Furthermore, the RECOMMENDER configuration was high-ranking in all four aspects: Informativeness (85.5%), preference management (80.0%), naturalness (79.3%), and coherence (84.1%).

The data for the configurations over the parameters were compared using a one-way analysis of variance (ANOVA)[2]. Preference management was perceived as significantly lower in the Q-A configuration compared to the other three configurations, where preferences indeed were modeled and de facto influenced the dialogue. PRYING received significantly lower ratings on coherence compared to the other three configurations. This is most likely due to that factual user queries were only used as indicators of preferences, and were not responded to in the way that configurations with delivery behaviors did. The RECOMMENDER configuration received a significantly higher rating on naturalness compared to the other three configurations.

The results show that BCORN's non-centralized approach that views dialogue strategy modeling as an emergent phenomenon is feasible, and encourages future development of the approach. They also imply that natural and coherent recommendation dialogue can be explained in terms of the suggested dialogue behaviors.

[2] $p < 0.001$ n.s. for all differences reported below.

References

Rodney A. Brooks. 1991. Intelligence without representation. *Artificial Intelligence*, 47:139–159.

Robin D. Burke, Kristian J. Hammond, and Benjamin C. Young. 1997. The findme approach to assisted browsing. *IEEE Expert*, 12(4):32–40.

Robin D. Burke. 2002. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*, 12:331–370.

Giuseppe Carenini, Jocelyin Smith, and David Poole. 2003. Towards More Conversational and Collaborative Recommender Systems. In *Proceedings of the International Conference of Intelligent User Interfaces*, pages 12–18, Miami, Florida, USA.

Cynthia Thompson, Mehmet Göker, and Pat Langley. 2004. A personalized system for conversational recommendations. *Journal of Artificial Intelligence Research*, 21:393–428.

Steve Whittaker and Marilyn Walker. 2004. Evaluating dialogue strategies in multimodal dialogue systems. In W. Minker, D. Bühler, and L. Dybkjaer, editors, *Spoken Multimodal Human-Computer Dialogue in Mobile Environments*, pages 247–268. Kluwer Academic Publishers.

Pontus Wärnestål, Lars Degerstedt, and Arne Jönsson. 2007. Interview and delivery: Dialogue strategies for conversational recommender systems. In *Proceedings of the 16th Nordic Conference of Computational Linguistics (Nodalida)*, Tartu, Estonia, May.

Pontus Wärnestål. 2005. User evaluation of a conversational recommender system. In Ingrid Zukerman, Jan Alexandersson, and Arne Jönsson, editors, *Proceedings of the 4th IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pages 32–39, Edinburgh, Scotland U.K.