A Constraint-Based Approach for Cooperative Information-Seeking Dialogue

Yan Qu
Clairvoyance Corporation
5001 Baum Boulevard, Suite 700
Pittsburgh, PA 15213
y.qu@clairvoyancecorp.com

Nancy Green
Department of Mathematical Sciences
University of North Carolina at Greensboro
Greensboro, NC 27402
nlgreen@uncg.edu

Abstract

We present a dialogue generation model, implemented in the COMIX prototype information system, which uses a Constraint-Based Problem-Solver (CBPS) to support cooperative mixed-initiative information-seeking dialogue. Use of the CBPS enables a dialogue system to 1) incrementally interleave query construction with solution construction 2) immediately detect under-constrained and over-constrained information requests, and 3) provide cooperative responses when these types of problems are detected. We also present a system evaluation investigating how well COMIX handles dialogues with over-constrained requests.

1 Introduction

Many existing dialogue systems adopt a two-phase approach to satisfying a user's request for information: query construction followed by solution construction and presentation, with the former concerned with the acquisition of the preferences and restrictions in a user's information needs, and the latter concerned with presenting solutions that satisfy those needs (e.g., Abella, et al., 1996; Litman, et al., 1998; Chu-Carroll, 2000). When we look at human-human information-seeking dialogues, however, we observe that a two-phase approach can lead to problems such as delayed identification of over-constrained problems (see example below).

A1: ok what flights do you want to have him on?
C2: ok he would like to be on United flight one one one seven
A3: first of all he's going all the way through to Copenhagen?
C4: oh, that's right
A5: and this is a connection in what city?
C6: ah Los Angeles
A7: and what time does that United flight leave?
C8: it's ah going out at uh two o'clock today, SFO to Los Angeles
A9: and then connecting to SAS flight nine thirty two?
C10: nine thirty two that's right
A11: actually the eleven seventeen I'm showing sold out

In this dialogue between a travel agent (A) and a customer (C) (SRI transcripts, 1992), the over-constraining attribute-value pair – flight United 1117 – occurred in turn C2, but was not detected until turn A11. Instead, the travel agent continues to collect more constraints to complete the information need specification only to find during the solution phase that the problem is over-constrained. Worse yet, if the human agent or information system cannot offer the user informed assistance, such as suggesting which constraint(s) to relax when an over-constrained situation is finally detected, the user is forced to adopt an inefficient trial-and-error strategy of selecting constraints to relax.

We present a low-level dialogue generation model that uses a Constraint-Based Problem-Solver (CBPS) to support cooperative mixed-initiative information-seeking dialogue. (We refer to the dialogue generation model as *low-level* since it does not address surface generation.) Use of the CBPS enables a dialogue system to 1) incrementally interleave query construction with solution construction 2) immediately detect under-constrained and over-constrained information requests, and 3) provide cooperative responses when these types of problems are detected. The model has been implemented in COMIX, a prototype system for providing airline flight information. In addition, we present a system evaluation designed to evaluate COMIX's performance when users make over-constrained requests.

2 Related Work

Cooperative and efficient dialogue between a user and information system has been an important goal in dialogue systems research for many years. Early work on cooperative response generation addressed how to generate responses explaining the reason for natural language query failures (Kaplan, 1979; Di Eugenio, 1987). However, those systems processed and responded to each query in isolation. Subsequent work addressed understanding the user's query in the context of the preceding discourse (Carbonell, 1983) and inferred plans (e.g., Allen and Perrault, 1980; Carberry, 1990). Raskutti and Zukerman (1997) describe a plan-based approach that integrates understanding the user's intentions with generating queries for clarification and generating responses to under-constrained and over-constrained requests. Their system does not use a constraint-based problem solver and does not provide access to data in a standard commercial database system.

Research on mixed-initiative dialogue has investigated techniques for deciding when a dialogue system should take initiative and techniques for content planning (e.g., see papers in Haller, Kobsa, and McRoy, 1999). Following Chu-Carroll and Brown (1999), we distinguish dialogue initiative from task initiative (i.e., taking the lead in problem-solving). In this view, a dialogue participant may take both types of initiative, dialogue initiative only, or neither. Preliminary studies suggest that human-computer dialogue is more efficient when a system dynamically allocates control of task initiative (Smith and Gordon, 1997). In COMIX, task initiative is exercised in asking for additional constraints when the user's request is under-constrained, or in suggesting constraints to relax when the request is over-constrained.

However, none of the above research has made use of a CBPS as a knowledge source for generating the system's contributions. Previous dialogue systems research involving constraint satisfaction models includes Jordan and Di Eugenio's (1997) analysis of cooperative problem-solving dialogue as a constraint satisfaction problem. However, they did not use the model to generate system responses. Donaldson and Cohen (1997) propose a constraint-based model to manage a system's turn taking goals, but not as a knowledge source for helping to meet a user's information needs.

3 Dialogue Generation Model of COMIX

In this section, we present our dialogue generation model as implemented in COMIX. First, we briefly describe the application domain and provide an example of user interaction with a version of COMIX (COMIX-MI) implemented for the system evaluation to be described in section 4. Then we present the COMIX system architecture and briefly describe the CBPS. Finally, we describe how the dialogue manager uses information provided by the CBPS to take initiative and to select dialogue actions.

3.1 COMIX application domain and COMIX-MI

COMIX queries a relational database of airline flight information for 8457 flights, created from data downloaded from a commercial database (www.nwa.com). (Although containing real world data, COMIX's relational database is static and was created for evaluation purposes only.) A form-based user interface, COMIX-MI ("Mixed Task Initiative"), was implemented for the experiment to be described in section 4. The experiment used form-based user interfaces in order to enable us to evaluate the performance of the system's low-level dialogue generation model independently of factors related to dialogue interpretation and surface generation. With COMIX-MI, the user specifies an information need by filling in fields on a query form. As soon as each field is filled in, COMIX attempts to update its interpretation of the user's query. If COMIX is unable to do so, e.g., because of an ambiguous value, the system takes dialogue initiative, e.g., by displaying a clarification dialogue window. After the user has finished filling in all non-optional fields and has requested COMIX-MI to search for flights, COMIX responds. If the request was over-constrained, then, using information provided by the CBPS, COMIX-MI takes the task initiative and suggests constraints to relax.

For example, the user interface of COMIX-MI is shown in Figure 1. The user has supplied values for attributes *departure city*, *arrival city* and *departure time* with preference strengths. Because the user's specification is over-constrained, COMIX-MI takes the task initiative to suggest which constraint to relax, as shown in Figure 2. It should be noted that the form-based user interface does not demonstrate all of the capabilities of COMIX described in this paper (such as providing suggestions for restricting under-constrained requests). Extensive simulation experiments to evaluate many of the capabilities of COMIX's dialogue generation model are described in (Qu, 2001).

3.2 COMIX architecture

Figure 3 shows the COMIX information system architecture. A user's input is first translated, e.g., from spoken or form-based input, into a semantic representation. The dialogue manager translates the semantic representation into constraints that are sent to the CBPS. (A constraint consists of an attribute-value pair and its preference strength ranging on a five-unit scale from Required to Weak). Also, the dialogue manager determines *when* it is licensed to take dialogue and task initiative and *what* dialogue action to perform. After a dialogue action has been selected, it is realized by the surface response generator; e.g., in COMIX-MI, the action to inform the user of the results of a successful query is realized as a results form.

Constraint-based problem solver

As shown in Figure 3, the CBPS has three subcomponents, labeled *Solution Construction*, *Solution Evaluation*, and *Solution Modification*, and maintains a solution synthesis graph (*SS-graph*). Nodes of the SS-graph represent subsets of user constraints and database tuples satisfying them. The Solution Construction module of the CBPS incrementally constructs and updates the SS-graph as user constraints are passed to it from the dialogue manager. Solution Construction queries the database (labeled *DBMS* in Figure 3) to obtain tuples satisfying the user's constraints. Solution Evaluation uses the current number of tuples satisfying the constraints in the SS-graph to determine if (1) a satisfactory solution has been found or (2) the problem is currently over-constrained or under-constrained. In case (2), Solution Modification uses the SS-graph and database domain constraints to identify relaxation or restriction candidates for over-constrained or under-constrained queries, respectively. Various solution modification heuristics have been evaluated through simulation experiments (Qu, 2001). For example, for over-constrained queries, Qu found that one effective heuristic is to identify constraints with weaker preference strengths that result in an empty solution set. COMIX has adopted this heuristic; thus, the system is able to suggest relaxation of the constraints that resulted in an over-constrained request but are less important to the user.

The solution set from the SS-graph and the relaxation or restriction candidates from Solution Modification are passed to the Dialogue Manager. The Dialogue Manager uses this information to select cooperative dialogue actions for resolving under-constrained or over-constrained situations as soon as they are detected. The cycle of interacting with the user to add or modify constraints can be repeated until a satisfactory solution is found. Details about the CBPS are presented in (Qu and Beale, 1999) and details about interaction with the database management system are given in (Qu, 2001).

Dialogue generation

COMIX extends the approach described in (Abella and Gorin, 1999) where a collection of dialogue motivators is iteratively tested and, whenever the test for a motivator succeeds, the motivator's associated dialogue action is performed. In COMIX, a dialogue motivator's test, or *applicability conditions*, may refer to information from the CBPS. Whereas in Abella and Gorin's approach, a motivator is associated with one action, in COMIX a motivator can be realized through one of several alternative dialogue actions; selection of the action depends upon information from the CBPS. Whenever at least one motivator applies, COMIX takes dialogue initiative.

The dialogue motivators and dialogue actions used in COMIX are based on an informal analysis of a corpus of naturally occurring information-seeking dialogues (SRI transcripts, 1992). The dialogue motivators, along with their applicability conditions, are shown in Table 1. For instance, the dialogue motivator Relaxation is invoked when the current specification of the user's information need is over-constrained (i.e., the solution set is empty) and relaxation of certain attributes could result in a non-empty solution set, while Restriction applies when the specification is under-constrained. In the form-based interfaces to COMIX used in the system evaluation, all motivators are tested in the order shown in Table 1 immediately after the user fills in a field in the query form. This order insures, e.g., that ambiguity (Clarification) is addressed before the system computes the results of the query (ProvideAnswer).

Table 2 presents the dialogue actions available to the dialogue manager for each motivator. Note that the dialogue actions associated with the Restriction and Relaxation motivators are performed only when the system is allowed to take task initiative in addition to dialogue initiative. In COMIX-MI, the system is allowed to take task initiative whenever one of these two motivators applies. For the Restriction and Relaxation motivators, a system allowed to take both task and dialogue initiative must decide among several actions. The choice is based upon information from the CBPS. For example, when Relaxation has been invoked and COMIX-MI must choose between ProposeNewVal (i.e., suggesting a relaxation candidate) and ProposeNewVal&Answer (i.e., providing a proposed relaxation candidate along with the resulting solution set of adopting the proposal), the latter action is preferred to the former when the resulting solution set would be small enough. Once a dialogue action has been selected, it is realized by the response generator, e.g., as form-based output in COMIX-MI.

4 System Evaluation

The purpose of the system evaluation is to evaluate whether the task initiative-taking actions enabled by the CBPS improve usability for over-constrained requests. We compare dialogues collected for two different task initiative settings through two versions of COMIX: COMIX-MI vs. COMIX-UI. COMIX-UI ("User Initiative"), is a version of COMIX with a form-based interface similar to the interface of COMIX-MI. The crucial difference between these two versions of COMIX is that COMIX-UI is *never* allowed to take task initiative while, as described in the preceding section, COMIX-MI is whenever certain motivators apply.

Initially, eight users were given flight reservation tasks that required them to access the airline schedule database described earlier. (The information needed to complete a task is referred to below as the *user's original information need*.) Each user participated in three sessions: a training session and two evaluation sessions, one with COMIX-MI and another with COMIX-UI. The order of working with COMIX-MI and COMIX-UI was varied to neutralize sequencing/learning effects. These eight users were each asked to solve four one-way trip tasks (two under-constrained and two over-constrained) and four round-trip tasks (again two under-constrained and two over-constrained). The users performed four tasks at each of the two evaluation sessions, i.e., the tasks were evenly distributed between the sessions with COMIX-UI and COMIX-MI. Due to the observed differences in the results for one-way and round-trip tasks, we recruited an additional eight users and repeated the above experiment for the four round-trip tasks only.

Data were collected in two ways: dialogue logs and user survey data. The system logged the total completion time of a dialogue (**Total Completion Time**), the user actions, the system actions, and the effect of these actions. From these logs, we compute the number of dialogue turns that the user takes (**User Turns**) and the number of dialogue turns the system takes (**System Turns**). Based on the user's recorded solution for each task, we calculate Kappa scores (Carletta, 1996) to measure the degree of task success (**Task Success**) with respect to the user's original information need. The user surveys, which measure subjective evaluation of the system by the user, were collected through a post-task questionnaire. The post-task questionnaire asks the user's judgment of task success (**User Task Success**) and user satisfaction. User satisfaction was calculated for each task from the user's response to a survey adapted from the questionnaire used by Walker et al. (1997).

We compared COMIX-MI and COMIX-UI with respect to the above features for two different types of tasks (finding one-way or round-trip flights) and for over-constrained situations.

4.1 Task complexity

For the one-way trip tasks, sixteen dialogues were recorded for each setting. We observed no statistically significant[1] differences between the two system settings for any of the performance measures. For the round-trip tasks, thirty-two dialogues were recorded for each setting. Table 3 summarizes the performance measures collected from the two system settings. COMIX-MI improves the dialogue efficiency measures by reducing the average total completion time by about 14%, the number of system turns by 35.2%, and the number of user turns by 49.6%. The improvements with these efficiency measures are statistically significant (*p=0.032* for **System Turns**, *p=0.020* for **User Turns**, and *p=0.015* for **Total Completion Time**). Task Success scores increase from 0.843 by the COMIX-UI setting to 0.889 by the COMIX-MI setting (*p=0.042*). The differences between the scores for **User Task Success** and **User Satisfaction**, however, are not significant.

To sum up, our experiments show that the system's task initiative-taking actions result in better performance in terms of dialogue efficiency and the Task Success scores for the harder round-trip tasks. For the easier one-way trip tasks, no significant differences are observed across the performance measures. User's perception of task success (i.e., **User Task Success**) and usability (i.e., **User Satisfaction**) is comparable for both system settings regardless of the system's task initiative-taking capability.

4.2 Over-constrained tasks

We further examined the effect of system initiative on the over-constrained tasks of the round-trip tasks. Sixteen dialogues were recorded for each setting.

Table 4 summarizes the performance measures collected from the two system settings. We observed that the COMIX-MI setting improves the dialogue efficiency measures by reducing the average total completion time by 18.6%, the number of system turns by 48.5%, and the number of user turns by 63.6%. The improvements with these efficiency measures are statistically significant (*p=0.031* for **System Turns**, *p=0.016* for **User Turns**, and *p=0.017* for **Total Completion Time**). A detailed analysis of the distributions of the numbers of system turns and users turns for the over-constrained round-trip tasks shows that the distributions of system turns and user turns are less spread out for COMIX-MI, and that the distributions of system turns and user turns are more spread out for COMIX-UI (Qu, 2001), which suggests that COMIX-UI is influenced to a greater degree by the variability in user's ability to find relaxation solutions, while COMIX-MI evens out the variability of user's problem-solving expertise by offering suggestions.

User satisfaction and the subjective task success scores (i.e., **User Task Success**) are very similar between COMIX-MI and COMIX-UI, as only one user gave up on one task. Task success as measured by Kappa, however, increased significantly by 13.5% from 0.693 by the COMIX-UI setting to 0.787 by the COMIX-MI setting (*p=0.035*). As there is not much difference in the subjective **User Task Success** scores, it suggests that user's perception of task completion/success is not always reliable, and does not correlate well with the objective Task Success scores.

In summary, our experiments show that the better performance we have observed with COMIX-MI in terms of dialogue efficiency and the Kappa scores for the round-trip tasks are associated primarily with the over-constrained round-trip tasks, for which COMIX-MI had the opportunities to offer cooperative relaxation suggestions. COMIX-MI also helps to even out the variability in user's ability to find relaxation candidates. User's perception of task success (i.e., **User Task Success**) and usability (i.e., **User Satisfaction**) is comparable for both system settings, but we have observed that user's perception of task success is not always reliable, and does not correlate well with the objective Task Success scores.

5 Conclusions and Future Work

This paper presents a constraint-based dialogue generation model to actively assist a user performing an information-seeking task. The CBPS of the model supports incremental problem formulation and solution construction and immediate detection of under- or over-constrained requests. The dialogue manager uses information provided by the CBPS to decide when to take task initiative and to

[1] Statistical significance in this work is measured with the paired sample *t* test, with the significance level at $\alpha<0.05$.

select cooperative dialogue actions. Results from the system evaluation have demonstrated that generation of cooperative dialogue action in over-constrained dialogue situations significantly improves dialogue efficiency and task success for more complex information seeking tasks such as round-trip flight reservation tasks. Possible future work includes performing a similar usability study for under-constrained requests, extending the approach to other domains (e.g., finding a hotel, buying a house or computer), and evaluating COMIX in spoken dialogue systems where noise from other system components such as the speech recognizer and parser may mitigate the effectiveness of COMIX.

References

Alicia Abella, Michael K. Brown, and Bruce Buntschuh. 1996. Development principles for dialogue-based interfaces. In *Proceedings of ECAI'96 Workshop on Dialogue Processing in Spoken Language Systems*, pages 1-7.

Alicia Abella and Allen L. Gorin. 1999. Construct algebra: analytical dialog management. In *Proceedings of the ACL'99*, pages 191-199.

James F. Allen and C. Raymond Perrault, 1980. Analyzing intention in utterances. *Artificial Intelligence*, 15:143-178.

Jaime J. Carbonell. 1983. Discourse pragmatics and ellipsis resolution in task-oriented natural language interfaces. In *Proceedings of the ACL'83*, pages 164-168.

Sandra Carberry. 1990. *Plan Recognition in Natural Language Dialogue*. ACL-MIT Press Series on Natural Language Processing. MIT Press, Cambridge, MA.

Jean C. Carletta. 1996. Accessing agreement on classification tasks: the kappa statistic. *Computational Linguistics*, 22(2):249-254.

Jennifer Chu-Carroll. 2000. MIMIC: An adaptive mixed initiative spoken dialogue system for information queries. In *Proceedings of the 6th Conference on Applied Natural Language Processing*, pages 97-104.

Jennifer Chu-Carroll and Michael K. Brown. 1999. An Evidential Model for Tracking Initiative in Collaborative Dialogue Interactions. In Susan Haller, Alfred Kobsa and Susan McRoy (eds).

Barbara Di Eugenio. 1987. Cooperative behavior in the FIDO system. *Information Systems*, 12(3):295-316.

Toby Donaldson and Robin Cohen. 1997. A constraint satisfaction framework for managing mixed-initiative discourse. In *Technical Reports of AAAI-97 Spring Symposium on Computational Models for Mixed-Initiative Interaction.*

Susan Haller, Alfred Kobsa and Susan McRoy (eds). 1999. *Computational Models of Mixed-Initiative Interaction.* Kluwer Academic Press, Dordrecht.

Pamela W. Jordan and Barbara Di Eugenio. 1997. Control and initiative in collaborative problem solving dialogues. In *Technical Reports of AAAI-97 Spring Symposium on Computational Models for Mixed-Initiative Interaction.*

Samuel Jerrold Kaplan. 1979. *Cooperative Responses from a Portable Natural Language Data Query System*. Ph.D. thesis, School of Computer and Information Science, University of Pennsylvania.

Diane J. Litman, Shimei Pan, and Marilyn A. Walker. 1998. Evaluating response strategies in a web-based spoken dialogue agent. In *Proceedings of ACL/COLING'98: 36th Annual Meeting of the Association of Computational Linguistics*, pages 780-786.

Bhavani Raskutti and Ingrid Zukerman. 1997. Generating Queries and Replies during Information-seeking Interactions. *International Journal of Human Computer Studies*, 47(6): 689-734.

Ronnie Smith and Steven Gordon. Effects of Variable Initiative on Linguistic Behavior in Human-Computer Spoken Natural Language Dialogue. *Computational Linguistics*, 23(1): 141-168.

Yan Qu. 2001. *A Constraint-Based Model of Mixed-Initiative Dialogue in Information-Seeking Interactions*. Ph.D. thesis, School of Computer Science, Carnegie Mellon University.

Yan Qu and Steve Beale. 1999. A constraint-based model for cooperative response generation in information dialogues. In *Proceedings of AAAI-99*, pages 148-155.

SRI Transcripts. 1992. Transcripts derived from audiotape conversations made at SRI International, Menlo Park, CA. Prepared by Jacqueline Kowtko under the direction of Patti Price.

Marilyn A. Walker, Diane Litman, Candace Kamn, and Alicia Abella. 1997. PARADISE: A general framework for evaluating spoken dialogue agents. In *Proceedings of the 35th Annual Meeting of the Association of Computational Linguistics*, pages 271-280.

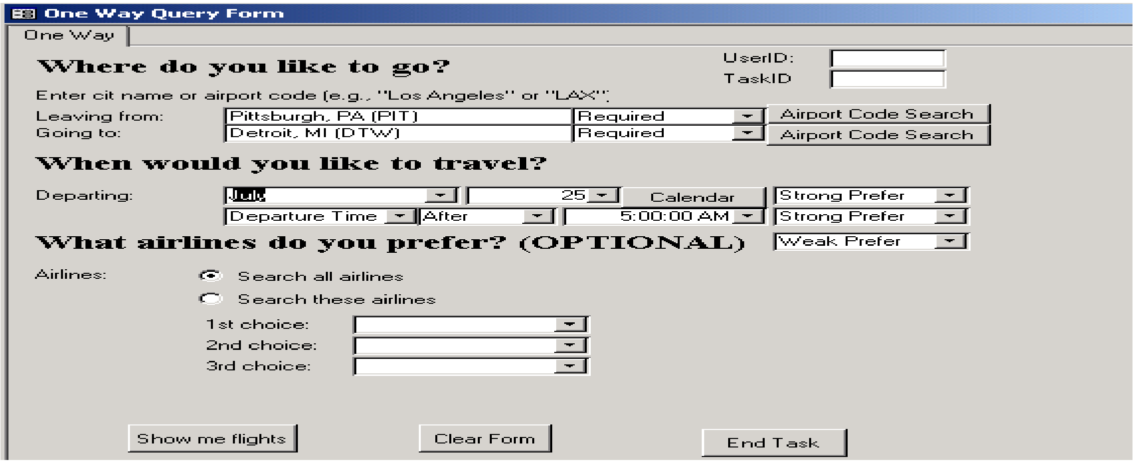

Figure 1: Flight query form in COMIX-MI.

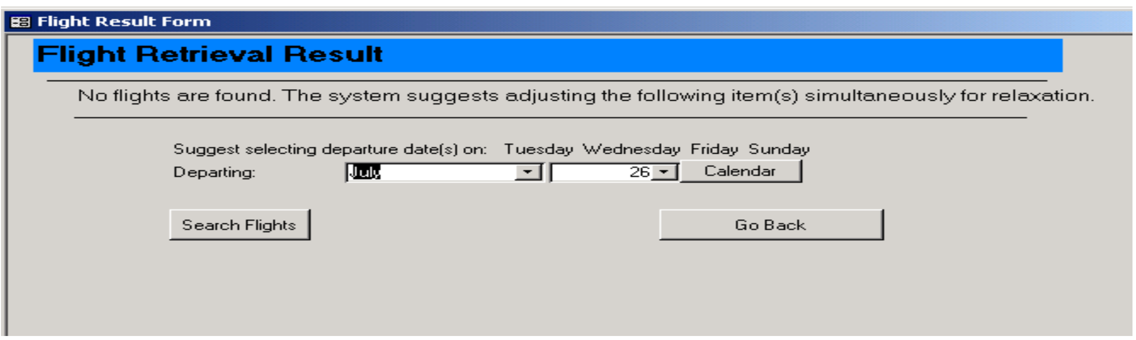

Figure 2: System response form for over-constrained queries in COMIX-MI.

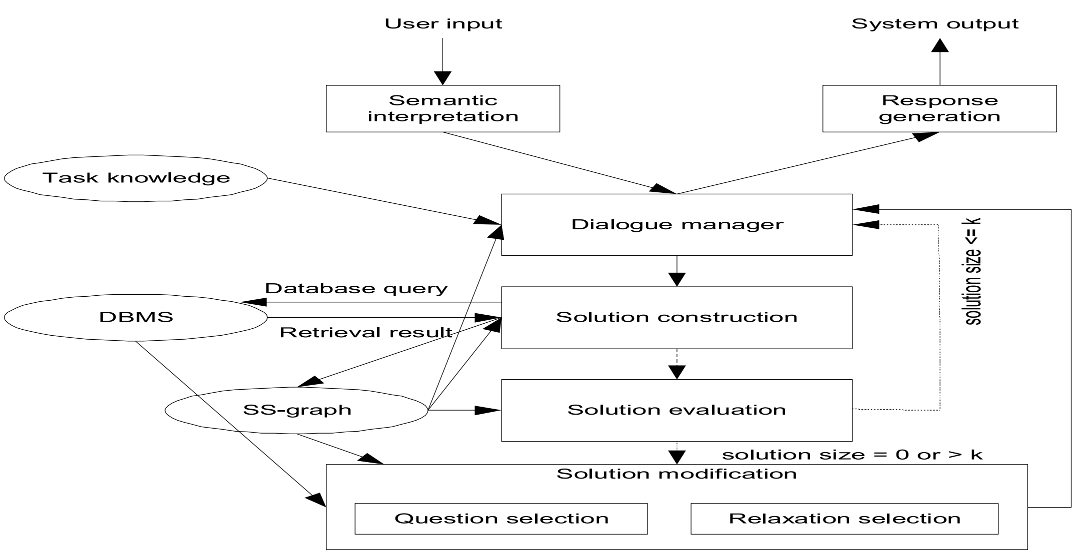

Figure 3: Architecture of COMIX. The rectangles represent modules and the ovals represent stored data. The solid lines with arrows represent data flow and the dashed lines with arrows represent control flow.

Dialogue motivators	Applicability conditions
ErrorCorrection	The value of an attribute in an object is invalid.
Clarification	An attribute in an object (other than the solution object) has more than one value.
ProvideAnswer	The value of the solution object is a non-empty set & The size of the non-empty set is equal to or smaller than a pre-determined size *k*.
NotifyFailure	The value of the solution object is an empty set.
Restriction	The value of the solution object is a non-empty set & The size of the non-empty set is greater than a pre-determined size *k* & There exist attributes in the object whose instantiation could result in a reduced set of solutions.
Relaxation	The value of the solution object is an empty set & There exist attributes in the object whose relaxation could result in an increased set of solutions.

Table 1: Summary of the applicability conditions for the dialogue motivators.

Dialogue Motivator	Task & Dialogue Initiative (COMIX-MI)	Dialogue Initiative only (COMIX-UI)
Restriction	RequestVal ProposeVal ProposeVal&Answer	No action
Relaxation	ProposeNewVal ProposeNewVal&Answer	No action
Clarification	Clarify	Clarify
ErrorCorrection	InformError	InformError
ProvideAnswer	Answer	Answer
NotifyFailure	InformFailure	InformFailure

Table 2: Selection of dialogue actions based on dialogue motivator and whether the system is allowed to take both dialogue and task initiative (column 2) or dialogue initiative only (column 3).

Performance measures	COMIX-UI	COMIX-MI	%change	p
System Turns	11.375	7.375	-35.2	*0.032*
User Turns	17.375	8.75	-49.6	*0.020*
Total Completion Time	381.313	328.094	-14.0	*0.015*
Task Success	0.843	0.889	+5.5	*0.042*
User Task Success	0.969	1	+3.2	0.163
User Satisfaction	33.469	33.719	+0.8	0.351

Table 3: Summary of round-trip results, with significant *p* values in italics.

Performance measures	COMIX-UI	COMIX-MI	%change	p
System Turns	16.5	8.5	-48.5	*0.031*
User Turns	27.625	10.063	-63.6	*0.016*
Total Completion Time	445.688	362.875	-18.6	*0.017*
Task Success	0.693	0.787	+13.5	*0.035*
User Task Success	0.938	1	+6.7	0.167
User Satisfaction	32.813	33.188	+1.2	0.382

Table 4: Summary of over-constrained round-trip results, with significant *p* values in italics.