Supporting Mixed Initiative Human-Robot Interaction: a Script-based Cognitive Architecture Approach

Hogun Park, Yoonjung Choi, Yuchul Jung, and Sung-Hyon Myaeng

Abstract— As complex indoor-robot systems are developed and deployed into the real-world, the demand for human-robot interaction is increasing. Mixed-initiative human-robot interaction is a good method to coordinate actions of a human and a robot in a complementary fashion. In order to support such interactions, we employ scripts that are rich, flexible, and extensible for a robot's interactions in a variety of situations. Scripts are amenable for expressing knowledge in an applicable form, especially describing a sequence of actions in organizing tasks. In this paper, we propose a script-based cognitive architecture for collaboration, which is based on three-level cognitive models. It incorporates Dynamic Bayesian Network (DBN) to automatically govern action sequences in the scripts and detect user's intention or goal. Starting from an understanding of user initiatives, our intelligent task manager suggests the most relevant initiatives for an efficient collaboration. DBN has been evaluated in real indoor task scenarios for its efficacy in interaction reduction, error minimization, and task satisfaction.

Index teams – Mixed-Initiative Interaction, Dynamic Bayesian Network, Human-Robot Interaction, Script, Robot-Task Script

I. Introduction

Human-Robot interaction (HRI) in the home-service domain requires a complex task that must be supported efficiently and effectively. Given the variety of situations a robot must face and users' goals and intentions that change dynamically, HRI is essential for successful task completion.

In order to improve the quality of interactions, many researchers have focused on mixed-initiative interaction (MII) [1][2] which was used in planning [7][8][9][12], intelligent agents [10][11], and HRI [13][15]. However, in MII, interaction (or collaboration) strategy to problem solving that users feel natural and efficient is still challenging because understanding user's intention has been rather troublesome. In this paper, we attempt to find user's intention through script-based cognitive architecture to suggest the most appropriate collaborative actions.

Scripts have several advantages in handling large-scale robot task knowledge in HRI [4][5]. First, scripts connect various pieces of existing knowledge in an applicable form. In other words, all agents, contexts, and predictable actions can be connected among each other. As a result, it is easy to see which actions are related to which contexts, and vice versa. Second, scripts are suitable for describing the sequence and the organization of tasks. This aspect is one of the attractive characteristics for collaboration between a user and a robot. A script is composed of a series of sub-tasks and their cause-effect relations. Using scripts, a robot can easily divide tasks and determine the actions for collaboration. For example, in the script that deals with a recipe, there are many sub-tasks such as cleaning and chopping vegetables. Since these sub-tasks are known to be a part of the work to be done, a robot can take an initiative for collaborative actions on the user's side, as in Fig 1. Our architecture utilizes these characteristics of scripts and helps a robot suggest appropriate actions for collaboration in accomplishing its tasks.

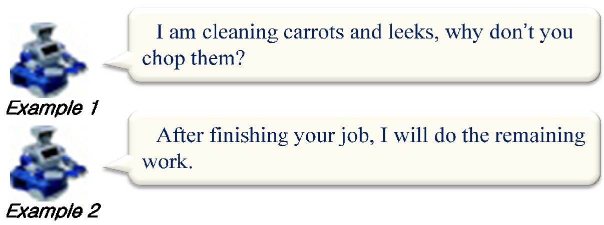

Figure 1. Interaction examples using scripts

Despite the advantages of scripts in HRI, a problem remains: how to model actions/objects in scripts and to utilize them to infer the intention of users along with incoming evidence connected to an action in a given task. Bayesian Network [6] can provide a probabilistic and decision-theoretic framework for an implementation. Although the framework is known to be effective under uncertain conditions [17][24], it is difficult to design such a network because it is often designed manually by experts as in most previous approaches [10][14][15].

In our approach, however, we propose a novel idea to model/learn a Bayesian network using scripts for efficient mixed-initiative interaction. In addition, we add a temporal dimension to a Bayesian network (henceforth, Dynamic Bayesian Network). In HRI, cause-effect is a very important

Manuscript received September 14, 2007. This research was performed for the intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Commerce, Industry and Energy of Korea, and (partially) supported by 2nd Phase of Brain Korea 21 project sponsored by Ministry of Education and Human Resources Development, Korea.

Hogun Park, Yoonjung Choi, Yuchul Jung, and Sung-Hyun Myaeng are with the School of Engineering, Information and Communications University, 119, Munji-ro, Yuseong-gu, Daejeon, 305-732, Republic of Korea (corresponding author to provide phone: 82-42-866-6210; fax: 82-42-866-6222; e-mail: gsgphg@ icu.ac.kr) {gsgphg, choiyj35, enthusia77, myaeng}@icu.ac.kr

relation that can be expressed by scripts where actions are sequentially related to other actions and contexts. For example, boiling water makes it possible to make coffee, and a broken boiler causes the task of finding thick clothes. Since cause-effect can be represented by an action transition along the time axis, our cognitive architecture utilizes DBN in solving uncertainty and inference problems.

In our previous work, cognitive architecture [5] and a case-based reasoning (CBR) approach [29] were proposed for task planning and adaptation, but it is limited to supervisory tasks. Even if a user asks for a specific task, a robot may have to respond with an initiation of a help and/or collaboration. In this paper, we describe our effort to design an interaction framework for mixed-initiative interaction that is more reliable in an unexpected situation. Our proposed approach was evaluated with about 200 hand-crafted scripts in a simulated home-service environment. The experiment showed a remarkable performance, 52.3% improvement in decreasing the number of interactions/errors and enhancing the user satisfaction, compared to existing goal inference methods for task scripts.

II. RELATED WORK

In many researches, Bayesian Network has been used in representing a variety of mixed-initiation interactions. Hong et al. [15] used Hierarchical Bayesian Network for prompting missing concepts and clarifying spurious concepts. In their model, user intentions were represented by many Goals and Mid-Goals. Goal inference was done by the two-level goal inferences, and if the probability of ongoing inference is not greater than a specific threshold, then it requests ambiguity solving to users.

Horvitz and Paek [10] employed a set of Bayesian Network to interpret the goals of speakers given evidences. Their model is organized in different levels of the task hierarchy. For progression to different levels, they performed cost-benefit analysis and made a decision. In the analysis, they considered two different asking costs and finally computed an expected cost.

Meng et al. [14] described mixed-initiative dialog modeling using Bayesian Network. They tried to govern the model transition for mixed-initiative interaction. Using pre-defined topology, they made a binary decision for inferring information goals of the user's query.

However, they designed their model in hand. Bayesian Network is very effective representation scheme, but the design of the network is very critical. In addition, they didn't consider the sequence of interaction over the network. The interaction is very relevant to the sequence of actions. Depending on which actions are done and which objects are handled, the next action can be different. To handle this sequence, we use Dynamic Bayesian network [16].

III. MIXED-INITIATIVE INTERACTION FRAMEWORK

A. Scenarios

The purpose of mixed-initiative interaction is to understand user's goal under uncertainty and to provide an appropriate interaction strategy. Before a robot's intervention, the user may be already doing some tasks. If the robot asks the user for information or to do something whenever uncertainty appears, he may feel bothered. The number of robot's requests should be minimized. In the following two sub-sections, two interaction scenarios are described.

1) A goal set up by the user

Assume that the robot has an obvious goal obtained from the human user. Based on this information, the robot selects a script that performs actions, which is most appropriate for achieving the goal. In this case, firstly, the user's verbal request must be transformed into our semantic frame of pre-defined script representation to be compared against pre-condition and post-condition of the scripts. Fig. 2 shows the process of this case. Using the cause-effect information of the scripts and scenes, the robot estimates relevance of the scripts [1]. After figuring out what tasks need to be accomplished, the robot may suggest collaborative actions. In this situation, there may be many collaboration strategies, depending on the degree of commitment to the task being solved and on the interruption strategy. Our three-level cognitive model (reactive, deliberative, and reflective models) supports the strategy using the scripts, world knowledge, and user models (Fig. 5).

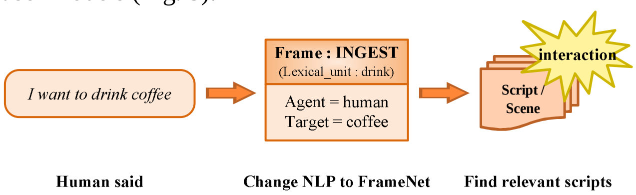

Fig 2. The process for finding relevant scripts using human user's goal

2) A goal inferred by the robot

The robot always has to be ready for observing users' actions and objects being manipulated. Through the task manager and the interaction manager (Fig. 5), it estimates which tasks are being done. If the probability of a task is not high enough to be the current task, the robot interacts with the user. The robot can get additional clues (explicit goal, actions, or contexts) for the user's goal through the interaction.

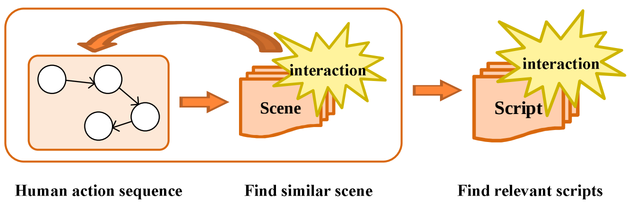

Fig 3. The process for finding similar scripts using robot's conjecture

[1] At this scenario, a task manager doesn't use DBN.

B. System Architecture

1) Interaction Manager

Our interaction starts from Interaction Manager (on lower right of Fig. 5) that observes users' actions or listens to his/her verbal requests. Whenever an input comes in, Interaction Manager executes dialogue/action analysis that generates a semantic frame of pre-defined script representation and delivers it to Task Manager (see Fig. 4 for an example).

```
<contents frame="ATRANS" lexical_units="have">
        <agent>User A</agent>
        <target>washtub</target>
</contents>
```

Fig. 4. An example semantic frame

After finishing a task, if it is under a new context or out of Script DB, Interaction Manager adds the solutions to Script DB. It is for solving a new problem based on the solutions of similar past problems.

2) Task Manager

Task Manager selects a service using the semantic frame provided by Interaction Manager. Selecting a service means that it identifies what users are doing and determines proper collaboration. In the process of service selection, the semantic frame becomes a state of Dynamic Bayesian Network (DBN), which is used for inferring users' objective (intent) hidden in actions. Semantic Frame Organizer supports the process of filling out the semantic frame which is not sufficiently complete. Our three-level cognitive model supports these two modules in Tasks Manager.

After service selection, Interaction Manager initiates collaboration with users. A strategy for the collaboration is determined by Deliberative Model's collaboration module. The detailed description of the process of inferring user intent and collaboration is found in Sections *IV*, *V*, and *VI*.

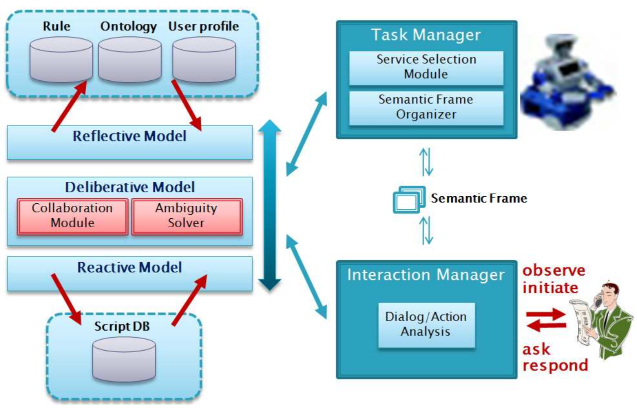

Fig. 5. System Architecture

3) Three-level Cognitive Models

The three models in our architecture are partially adopted from the EM-ONE architecture [4]. The Reactive Model interacts with a knowledge base of scripts that may be of various lengths and sizes. It helps finding relevant actions from the knowledge base, compared with the selected semantic frame. If it cannot find relevant actions or it fails to infer the intent of the user, Ambiguity Solver in Deliberative Model tries to interact with Reflective Model or users. Reflective Model utilizes existing rules, ontology, and the user profile to resolve the ambiguity, and returns a result. If Service Selection Module in Task Manager succeeds in inferring the intent, Collaboration Modules (Sections *VI*) tries to select the proper initiation.

IV. Script Structure

We utilize a robot task script structure developed by the previous work [5] but in a modified form to make it more machine-readable and semantically less ambiguous. A natural language was used for representing actions, corresponding properties, and cause-effects in the previous work. Because of the difficulties in understanding their semantics and comparing them, however, we chose a structure like FrameNet [18]. To represent frequently used robot's actions, we chose seven action frames (*ATRANS*, *PTRANS*, *PROPEL*, *GRASP*, *INGEST*, *MTRANS*, and *MUBILD*) adopted from the Conceptual Dependency (CD) theory [19], where a set of primitive actions was defined. Detailed elements for the chosen frames are shown in Fig 6.

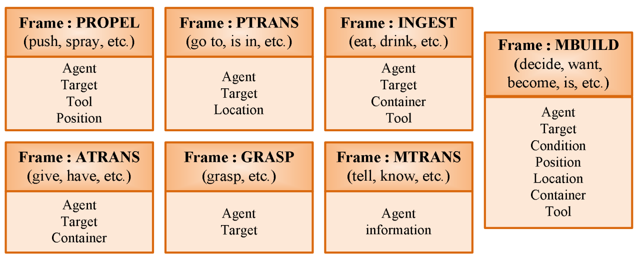

Fig 6. The frame elements for seven frames

Each action frame has a number of primitive actions (from [25]), which are also represented in a more machine-readable and clearly understandable format using pre-defined primitive action frames that contain properties such as object, verb, container, tool, currentPosition, nextPosition, and so on. An Example of our script can be found in (http://ir.icu.ac.kr/~phg/idro/script_example1.xml).

V. Script Selection System

A. Representing robot task scripts with DBN

Every task has a set of action sequences. For example, before executing an "*open the refrigerator*" action, it must perform a "*go to the refrigerator*" action. In order to represent this sequence and cause-effect relations between actions, our system chooses the dynamic Bayesian network (DBN) method that models time series [20].

Our scripts are represented in DBN (Fig. 7) that has three parts: an action sequence part, a scene part, and an object/environment part. In action sequence part, each node representing an action is based on observations. The actions are sequentially related, and a sequence of actions organizes a scene (a unit of the temporal aspect of actions). This hierarchical DBN module is for higher level context reasoning. This idea has an advantage for integrating action-level reasoning and domain knowledge. In a scene part, all possible nodes are composed of scenes from script knowledge base. Thus, each scene data is from previous action history that has accumulated through experience. An object/environment part is intended to represent how actions are related to context. In this representation, this part regards handled objects which are shown in scripts as a node.

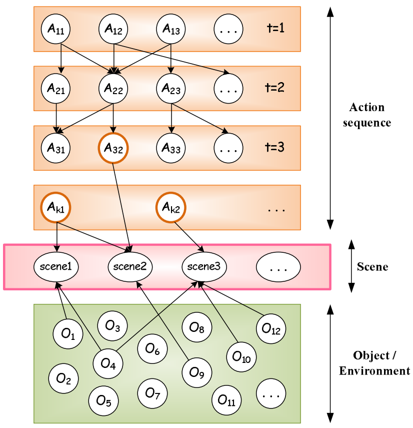

Fig 7. The action sequence represented as DBN

In the action sequence part, a number of time sequences are represented. At time *t*, our system observes semantic frames generated from dialog/action analysis and selects corresponding inferred relevant actions[2] (A_{t1}, A_{t2}, …, A_{tn}) from Script DB. Thus, the actions at the same level are to be executed together. Actions that are larger than a specific similarity value are selected as candidates and the similarities are assigned to the probabilities of the actions. The similarity value is obtained by calculating how similar the elements contained are[3]. The elements are a set of frame attributes, as in Fig 6. Each action causes other actions at the next step, establishing a dependency relation. In Fig. 7, A_{11} has dependency on A_{21} and A_{22}. Based on this information, a robot can estimate which scene in the existing script knowledge base is most strongly related. The computation is described in the next subsection.

This likelihood is also affected from the bottom part (object/environment part) where scripts' context information is described. Therefore, our estimation is obtained from two Bayesian networks based on the action sequence and context information.

[2] Actions can be exploited from each script (see our robot task script structure.[5])

[3] $\sum_i \sum_j S(a_i, a_j)$ / (# of comparisons), where function S returns 0 or 1 (both are same), and a_i & a_j are attributes from action semantic frames asf_1 and asf_2 respectively.

1) Setting initial values and updating

Each state (in this paper, action) has a state transition probability table that includes all histories, the current state, and next possible states. The probability is proportional to (1).

$$W_s = \sum_{d=1}^{d=|D|} |F_{sd}| \cdot log\,(1 + \frac{N_d}{n_{sd}}) \qquad (1)$$

$$|F_{sd}| = \frac{F_{sd}}{\sqrt{\sum_{k=1}^{k=K} F_{kd}^2}}$$

where W_s is the weight of state transition s,
F_{sd} is the frequency of state transition *s* in domain *d*,
n_{sd} is the number of documents in domain *d* where state transition *s* occurs,
N_D is the total number of scripts in domain *D*,
K is the total number of terms in domain *d*, and
$|D|$ is the number of domains.

This probability assignment is similar to TF-IDF [3]. In our weighting scheme, it emphasizes the importance of each state transition and assigns a higher weight to a state transition when it is reached with a high frequency and a low script frequency of the transition in the whole collection of scripts. The difference between TF-IDF and W_S is that we summarize individual TF-IDF values in each domain[4]. It needs a lot of space for saving all available action transitions. However, our model doesn't care about space complexity. Our DBN model has a small number of time slices because the most scenes have less than five actions. Although we have a lot of actions, next predictable actions are restricted to causal actions. Therefore, space complexity is a not a serious problem.

Most DBN approaches need some kind of learning, due to missing beliefs in the networks [27]. Generally, the expectation-maximization (EM) algorithm is used to cope with missing beliefs [26][27][28]. However, in our model, we assume full observability condition; the values of all states are known (i.e. all states are observable).

2) Definition of the conditional probability distribution

As we described, our model has different conditional probability tables, compared to the original Bayesian network. In the original one, it considers only previous states for calculating conditional probabilities [21]. In our model, however, it must consider all trajectories of the process because it requires action histories for determination of most likely one. Therefore, the conditional probability distribution in the action sequence parts can be written as follows:

$$P\,(\mathbb{A}(t+1) \mid \mathbb{A}(1), \cdots, \mathbb{A}(t)) \qquad (2)$$

where $\mathbb{A}(t)$ is a certain action at time *t*. The conditional probability distribution for determining a scene is similar:

$$P\,(\mathbb{S} \mid \mathbb{A}(1), \cdots, \mathbb{A}(T)) \qquad (3)$$

where $\mathbb{S}$ is a certain scene and *T* is the number of actions.

This distribution can be extremely complex. In the worst case, each action is connected with all actions in the next time slices. As a result, the time complexity is $O(n^{t-1})$ where *n* is the number of actions and *t* is the number of time slices,

[4] In our implementation, this domain means locations such as a room and a kitchen.

which is the reason why almost all papers assume that the process is *Markovian* [22]. However, our model may not have high complexity because the values of *n* (as in *Section IV*) and *t* are small.

B. Script Selection

One script can have several scenes, and one scene can belong to one or more scripts. If it knows one or more scenes, it can guess similar scripts that contain these scenes. Through inferring one or more scenes, we can estimate which script is similar. Using similarities between estimated potential scenes and scripts, the best relevant script is selected.

VI. Collaboration Strategy

Our cognitive architecture includes the process of determining a collaboration strategy in the middle of HRI. If there are dominant rules or user profiles[5], it suggests a corresponding collaboration. For example, if a user is handicapped, the strategy for uncomfortable users will be selected. In general, for selecting a collaboration strategy, Task Manager uses mutual information[6] *I(X; Y)*,

$$I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$$
$$= \sum_{x,y} p(x,y) \log \frac{p(x|y)}{p(x)} \qquad (4)$$

where $x, y \in S$, S is a set of scenes of a specific script, and *p(x)* is the probability about the scene *x* is likely to occur in all script spaces.

If two scenes are independent, the dependency between two scenes is smaller than a threshold. A collaboration module calculates this mutual information for all the pairs. It can suggest a collaboration strategy that minimizes the mutual information among tasks of each participant.

VII. Experiments

A. Environment Setting

For evaluation, we have designed a virtual home-service environment which was implemented in graphical user interface. The service space is composed of seven places (a bed room, a toilet, a library, a living room, a kitchen, an entrance, and a multipurpose room.) Through the user interface, users take an action, interact with a robot, and get visual feedback.

[5] This knowledge can be obtained from a reflective model.

[6] Collaboration Module in the deliberative model plays the role of calculating this mutual information.

B. Experimental Design

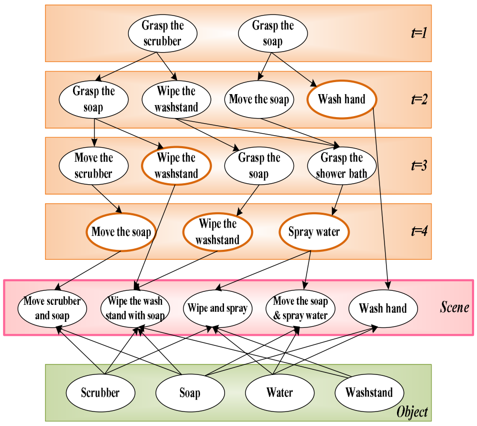

Fig 8. An example DBN

For our experiments, we made example tasks that are frequently used in the task sequence corpus of OMICS[7]. OMICS is an attempt to make indoor mobile robots smarter, and they have collected several thousands of pieces of ordinary knowledge that constitute "common-sense" for an indoor home-service. Based on the selected task sequence knowledge in OMICS, we tested our approach in three different scenarios: one is a scenario where the user sets up a goal (Explicit Scenario, one of two basic scenarios in *Section III. A*,); the second one is a scenario that a robot infers a goal (Ambiguous Scenario, another basic scenario in *Section III. B*,), and the last one is a scenario containing one irrelevant user action (Noisy Scenario.) Last two scenarios are to test how much each method is reliable. Following the scenarios, participants performed the tasks, waiting for initiation and collaboration of the robot. Participants were divided into three groups and performed each scenario.

	Description
Explicit Scenario	Tell a goal to the robot : clean a floor Action 1: get a broom Action 2: sweep the floor Action 3 put away the broom Action 4: take out a mop Action 5: take out the mop bucket Action 6: place water in the mop bucket Action 7: dip the mop in the bucket Action 8: run the mop over the floor Action 9: put mop away when the floor is clean
Ambiguous Scenario	Do not Tell a goal to robot Action 1: get a broom Action 2: sweep the floor Action 3 put away the broom Action 4: take out a mop Action 5: take out the mop bucket Action 6: place water in the mop bucket Action 7: dip the mop in the bucket Action 8: run the mop over the floor Action 9: put mop away when the floor is clean
Noisy	Tell a goal to the robot : clean the floor

[7] http://openmind.hri-us.com

Scenario	
	Action 1: get a broom Action 2: sweep the floor (Noisy Action 1): drink a water Action 3 put away the broom Action 4: take out a mop Action 5: take out the mop bucket Action 6: place water in the mop bucket Action 7: dip the mop in the bucket Action 8: run the mop over the floor Action 9: put mop away when the floor is clean

Fig 9. Three scenarios

In out experiments, we compared our model with three different models, as in Table I. The first one is TF-IDF that does not employ a temporal factor. This method is compared with other methods that have dependency. The second is also TF-IDF, but it additionally considers action sequences. That is, it counts only scenes that have two continuous action sequences. The last one is a Bayesian network method. We utilized [15]'s suggestion to represent a Bayesian network.

TABLE I
EXPERIMENTAL METHODS

	Temporal Factor	No Temporal Factor
Dependency	Our method	Bayesian Network [15]
No dependency	TF/IDF with action sequence	TF/IDF

These four methods infer users' intent for service selection[8] on our Cognitive framework, using our 200 manually crafted scripts[9]. Before experiments, all prepossessing (e.g. indexing and computing conditional probability tables) were done.

Here is the hypothesis of our experiments:

** Hypothesis: Compared to conventional interactions method for service selection, our proposed system is efficient with respect to the number of errors, the number of interactions, and task satisfaction ratio.*

The interaction appears when there is not enough probability to estimate an appropriate service or the service selection was incorrect. Therefore, if each method cannot find promising scenes, interaction[10] can occur. The error happens when an incorrect suggestion is made. Since this is originated from wrong inferences, it is an important factor. Task Satisfaction is to measure how many intended goals are accomplished within a certain time unit.

C. Experiment Results

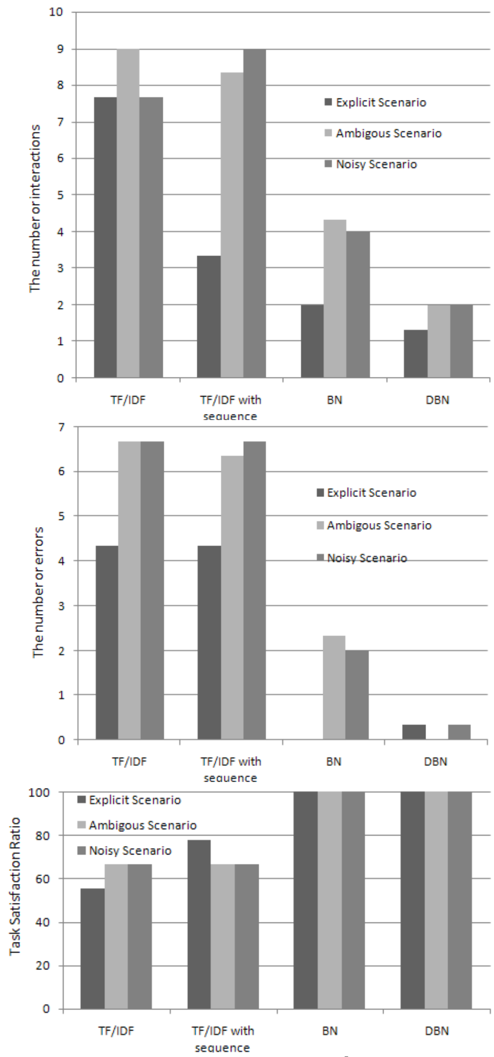

Fig. 10 Experiment results

Fig. 10 shows the result of our experiments in terms of the number of interactions. Compared with the first two methods (TF/IDF and TF/IDF with sequences), the last two (BN & DBN) showed improvements. Because TF/IDF does not consider dependency of actions and contexts, overall performance was low. Between BN and DBN, there are slight differences. In case of the explicit scenario, both showed similar results. However, in the remaining cases, DBN performed better because Bayesian Network with the time dimension is more effective in solving an ambiguous and noisy situation.

In the experiment for the number of errors, BN and DBN showed extremely high performance. This is largely attributed to dependency representation. Also, as in the previous results, DBN was also better in dealing with ambiguous and noisy scenario.

The Task Satisfaction experiment focuses on a measurement about how many intended goals are accomplished within a certain time. BN and DBN finished their missions in a shorter time than TF/IDF and TF/IDF with sequences.

[8] Except the service selection, the remaining parts were all same.

[9] These home service scripts were built by two domain experts. The examples of home-service tasks are "*bring me something,*" "*clean the room,*" and so on.

[10] The number of interactions is restricted to at most 10.

Table II shows the performance differences, compared to TD/IDF. While all the three methods showed good performance, DBN showed the best results, proving that it is a powerful inferring method in script-based architecture.

TABLE II
PERFORMANCE COMPARISON

	TD/IDF with sequence	BN	DBN
# of interactions	↓12.2%	↓46.6%	↓63.3%
# of errors	↓11.1%	↓44.4%	↓56.6%
Task Satisfaction Ratio	↑7.44%	↑37.1%	↑37.1%

VIII. CONCLUSION

In this paper, we proposed a script-based cognitive mixed-initiative interaction framework for collaboration. Our new interaction architecture understands the goal of users and helps task allocations. Three cognitive models (reactive, deliberative, and reflective models) of our framework support inference of user's intent and selection of a collaboration strategy. In inferring user's intent, Dynamic Bayesian Network was used. Different from existing possible methods (TF/IDF and [15]) it considers casual-effect information along with the time axis script estimation. The adoption of DBN has contributed to minimizing the number of interactions and errors, and increasing the task satisfaction, and the enhancement is about 52.3 % on average.

In our framework, many qualitative and quantitative scripts are needed. For further extension to other areas, we need automatic script-story generation like automatic story-telling approaches. Future research may take a more dynamic environment with various scripts and develop a flexible action sequence adaptation module for robust goal completion.

ACKNOWLEDGMENT

This work was supported by Korea Foundation for International Cooperation of Science & Technology (KICOS) through a grant provided by the Korean Ministry of Science & Technology (MOST) in K20711000007-07A0100-00710, and supported by the Intelligent Robotics Development Program, one of the 21 Frontier R&D Programs funded by the Ministry of Commerce, Industry and Energy of Korea.

REFERENCES

[1] J. Allen, "Mixed Initiative Interaction," IEEE Intelligent Systems, vol. 15, no. 4, pp. 14-23, 1999.

[2] E. Horvitz, "Uncertainty, action, and interaction: In pursuit of mixedinitiative computing," IEEE Intelligent Systems, vol. 14, no. 5, pp. 17-20, 1999.

[3] G. Salton and C.S. Yang, "On the Specification of Term Values in Automatic Indexing," J. Documentation, pp. 351-372, 1973.

[4] P. Singh, EM-ONE: An Architecture for Reflective Commonsense Thinking, Ph.D. thesis, Computer Science and Engineering at MIT, 2005.

[5] Y. Jung, Y. Choi, H. Park, W. Shin and S. H. Myaeng, "Integrating Robot Task Scripts with a Cognitive Architecture for Cognitive Human-Robot Interactions," in *2007 IEEE Int. Conf. Information Reuse and Integration (IRI07)*, pp. 152-157.

[6] J. Pearl, "Probabilistic reasoning in intelligent system: Networks of plausible inference," *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Pub., Inc, 1988.

[7] M. Burnstein, G. Ferguson, and J. Allen, "Integrating Agent-Based Mixed-Initiative Control with an Existing Multi-Agent Planning System," University of Rochester, Computer Science Department, Technical Report 729, 2003.

[8] M. A. Hearst, Trends & Controversies: Mixed-Initiative Interaction. IEEE Intelligent Systems, vol. 14, no. 5, pp. 14-23, 1999.

[9] S. Mitchell, 1997. An Architecture for Real-Time Mixed Initiative Collaborative Planning. *1997 AAAI Spring Symposium: Computational Models for Mixed Initiative Interaction*, Tech. Rep. SS-97-04.

[10] E. Horvitz and T. Paek, "A computational architecture for conversation," *Proc. of the 7th Int. Conf. on User Modeling*, pp.201- 210, 1999.

[11] S. Han, S. Cho, "A hybrid personal assistant based on Bayesian networks and a rule-based system inside a smartphone," *Int. J. Hybrid Intell. Syst*, vol. 2, no. 3, pp. 221-234, 2005.

[12] M. T. Cox, "Planning as mixed-initiative goal manipulation," In *Proc. of the Workshop on Mixed-Initiative Intelligent Systems at the 18th Int. Conf. on Artificial Intelligence*. Menlo Park, CA: AAAI Press, 2003.

[13] B. Bell, J. Franke, H. Mendenhall, "Leveraging Task Models for Team Intent Inference," In *Proc. Int'l Conference on A I*, 2000

[14] H. Meng, C. Wai and R. Pieraccini, "The use of belief networks for mixed-initiative dialog modeling," *IEEE Trans. On Speech and Audio Processing*, vol. 11, no.6, pp. 757-773, 2003.

[15] J. H. Hong, Y. S. Song, and S. B. Cho, "A Hierarchical Bayesian Network for Mixed-Initiative Human-Robot Interaction," in *Proc. 2005 IEEE Int. Conf. on Robotics and Automation (ICRA 2005)*, pp. 3808-3813, 2005.

[16] G. Provan, "Tradeoffs in constructing and evaluating temporal influence diagrams," *Proc. Uncertainty in AI*, pp. 40-47, 1993

[17] J. H. Hong and S. B. Cho, "A two-stage Bayesian Network for effective development of conversational agent," LNCS 2690, pp. 1-8, 2003.

[18] C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The Berkeley FrameNet project," in *Proc. 36th Annu. The Association for Computational Linguistics and 17th Int. Conf. on Computational Linguistics (ACL-COLING)*, pp. 86-90, 1998.

[19] R. C. Schank and R. P. Abelson, (1977, July, 1), *Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structure*.

[20] A. Turcker and X. Liu, "Learning Dynamic Bayesian Networks from Multivariate Time Series with Changing Dependencies," in *Proc. 5th Int. Sym. on Intelligent Data Analysis Conference*, pp. 100-110, 2003.

[21] S. B. Im, K. S. Hwang, and S. B. Cho, "A Bayesian Network Framework for Vision Based Semantic Scene Understanding," in *Proc 16th Int. Conf. on Robot & Human Interactive Communication (RO-MAN 2007)*, pp. 839-844, 2007.

[22] N. Friedman, K. Murphy, and S. Russel, "Learning the Structure of Dynamic Probabilistic Networks," in *Proc 14th Annu. Conf. on Uncertainty in Artificial Intelligence (UAI-98)*, pp. 139-147, 1998.

[23] D. Barber, "Dynamic Bayesian Networks with Deterministic Latent Tables," in *Proc.6th Annu. Conf. on Neural Information Processing Systems (NIPS 2002)*, pp.713-720, 2002.

[24] P. Korpipää, M. Koskinen, J. Peltola, S. M. Mäkelä1 and T. Seppänen, "Bayesian approach to sensor-based context awareness," Personal and Ubiquitous Computing, vol. 7, pp. 113-124, 2003.

[25] H. S. Kim, Y. C. Jung, and Y. K. Hwang, "Taxonomy of Atomic Actions for Home-Service Robots", J. of Advanced Computational Intelligence and Intelligence Informatics (JACIII), pp. 114-120, 2005.

[26] Y. Du, F. Chen, W. Xu, and Y. Li, "Recognizing Interaction Activities using Dynamic Bayesian Network," in *Proc. 18th Int. Conf. on Pattern Recognition (ICPR 06)*, pp. 618-621, 2006

[27] Y. Zhang, Z. Deng, H. Jiand, and P. Jia, "Dynamic Bayesian Network (DBN) with Structure Expectation Maximization (SEM) for Modeling of Gene Network from Time Series Gene Expression Data," in *Proc 2006 Int. Conf. on Bioinformatics & Computational Biology (BIOCOMP 06)*, pp. 41-47.

[28] V. Mihajlovic and M. Petkovic, "Dynamic Bayesian Networks: A State of the Art," Centre for Telematics and Information Technology, University of Twente, Netherlands, Tech. Rep., TR-CTIT-01-35, 2001.

[29] Y. Jung, H. Park, Y. Choi, S. H. Myaeng, "Designing a Cognitive Case-Based Reasoning Framework," in *Proc. 16th IEEE International Symposium on Robot & Human Interactive Communication (ROMAN 07)*, 2007.