Intention Recognition using a Graph Representation

So-Jeong Youn, and Kyung-Whan Oh

***Abstract*—**The human friendly interaction is the key function of a human-centered system. Over the years, it has received much attention to develop the convenient interaction through intention recognition. Intention recognition processes multimodal inputs including speech, face images, and body gestures. In this paper, we suggest a novel approach of intention recognition using a graph representation called *Intention Graph*. A concept of *valid intention* is proposed, as a target of intention recognition. Our approach has two phases: goal recognition phase and intention recognition phase. In the goal recognition phase, we generate an action graph based on the observed actions, and then the candidate goals and their plans are recognized. In the intention recognition phase, the intention is recognized with relevant goals and user profile. We show that the algorithm has polynomial time complexity. The intention graph is applied to a simple briefcase domain to test our model.

***Keywords*—**Intention recognition, intention, graph, HCI.

I. INTRODUCTION

THE design of a human-friendly system is a goal of Human-Computer Interaction (HCI) or Human-Robot Interaction (HRI). Many researchers make efforts to develop the convenient interaction providing natural language processing, voice recognition, and gesture recognition.

Recently, intention modeling and recognition are important research issues in HCI and HRI [1]. It is very important because the systems can not support human adequately without knowing what the human wants to be done. Human can inform the system of his intention by text or speech explicitly. Also he can do it implicitly by doing something related his intention. It is easy for the system to understand the explicitly represented intentions like “copy this file” in HCI, or “open the window” in HRI. On the contrary, implicitly represented intentions might not be clear to the system. There have been many researches to handle this problem. We focus on the intention recognition by observing human behavior.

Intention modeling is an interesting research area and common issue to psychology and cognitive science. Some researches of computer science and robotics have shown good results by using the fruit of cognitive science, and psychology. One of them is [2]. They used mental model of [3] and intent signal decomposition of [4] to suggest an intention reading model. They formulated an intention reading problem as a function of actions, tasks, and a psysico-mental state. The intention model in [2] is shown in Fig. 1. In [2], an intention has the same meaning as a goal.

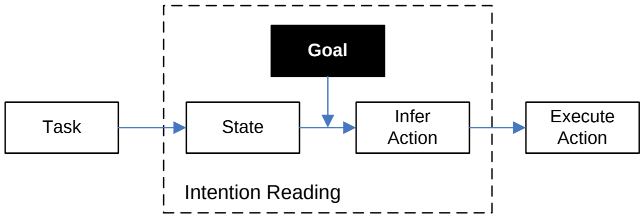

Fig. 1 A Model of Intention Reading [2]

As we can see in [2], an intention and a goal are used in the same way in an intention or a goal recognition problem. A goal is usually a conjunction of subgoals, and has a hierarchical structure. There are some ambiguities interpreting what is the final target or goal when a system recognizes a goal. Is this enough to describe user intention? Or is there another goal which is in deeper abstraction level? Therefore, we decide to use the term *goal* and *intention* in different meaning. A *goal* is something that a human hopes to achieve. That is, a goal is the desired state of affairs of a human and is the result of a sequence of actions. An *intention* is an idea or a mental state of what a human is going to do. If a man has in mind to quit smoking, that is an intention. But if he decides to quit smoking to change himself in the New Year, that is a goal. After he makes an action plan, he can achieve the goal doing actions sequentially. This process is shown in Fig. 2.

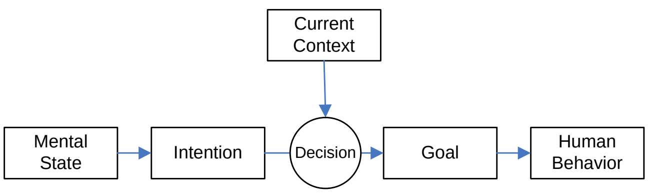

Fig. 2 Generation of Human Behavior

Intention recognition is a reverse process of the behavior generation. At first, human actions are observed. Then, a goal can be recognized through observed actions. With the achieved goal, we can recognize human intention under context.

Manuscript received November 30, 2006. This work was supported in part for the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs by the Ministry of Commerce, Industry and Energy of Korea.

So-Jeong Youn is with the Department of Computer Science, Sogang University, Seoul, Korea (phone: +82-2-703-7626; fax: +82-2-703-7626; e-mail: sjyoun68@naver.com).

Kyung-Whan Oh is with the Department of Computer Science, Sogang University, Seoul, Korea (e-mail: kwoh@sogang.ac.kr).

Open Science Index, Computer and Information Engineering Vol:1, No:1, 2007 waset.org/Publication/11699

Open Science Index, Computer and Information Engineering Vol:1, No:1, 2007 waset.org/Publication/11699

In this paper, we suggest a method of recognizing intention by observing user's behavior, finding relevant goals, and considering current context. In this method, we represent the relations among the intension, goal, and actions as a graph to recognize intention. We call this representation *Intention Graph*.

Intention graph is inspired by Goal Graph in [5] and Graphplan in [6]. Blum suggests a new approach to planning based on compact structure, Graphplan. Jun Hong improves it to recognize fully and partially achieved goals and apply it to large scale Unix domain which has 100,000 goals. We improve Jun Hong's Goal Graph to recognize intentions using recognized goals and user profile, and apply it to modified briefcase domain.

The structure of this paper is as follows: Section II defines an Intention Graph and few concepts used in our graph. In section III, five algorithms are suggested to recognize intentions based on Intention Graph. Section IV shows a briefcase domain with Intention Graph. In this domain, we define some goals, intentions, and user profile information. We will give a brief conclusion in section V.

II. Intention Graph

A. Organization of Intention Graph

Intention graph consists of state, action, goal, and intention nodes and edges. It is represented as $IG = <S, A, G, I, E>$ where S is a set of state node, A is a set of action node, G is a set of goal node, I is a set of intention node, and E is a set of edges.

S_t is a state set at time step t. Each state node represents a ground literal which values are True. The negative literal ¬P can be used as a state. The closed-world assumption is used, meaning that any conditions that are not mentioned in a state are assumed false. A special subset of S is a set of initial states and is denoted as S_0. We assume that the initial states are given completely.

An instance of action schema consists of a set of preconditions and a set of effects. A precondition set is a conjunction of positive literals stating what must be true in a state before the action can be executed. An effect set is a conjunction of literals describing how the state changes when the action is executed.

An instance of a goal schema consists of desired states, and they are called *goal descriptions*. An instance of an intention schema consists of ground goal conditions and related user profile information. Each edge represents the relations between nodes. An example of intention graph is shown in Fig. 3.

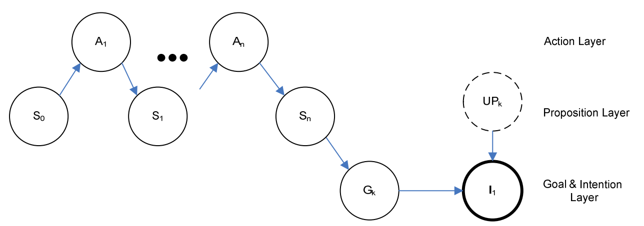

Fig. 3 An Example of Intention Graph

Intention graph has three layers: Action layer has action nodes, proposition layer has state nodes including states for user profile information, and goal & intention layer has recognized goal and intention nodes. There is one action node in each time step. New time step starts when an action is observed.

A state node at time t is represented as *state(s,t)* where s is a ground literal. The initial state is *state(s,0)*. An action node is represented by *action(a,t)* where a is an observed action at time t. A goal node is represented by *goal(g,t)* where g is a goal recognized at time t. An intention node is represented by *intent(i,t)* where i is an intention. There are six kinds of edges in intention graph. A precondition edge connects an action node with its precondition state node and is represented by *precondition-edge(state(s,t),action(a,t+1))*. An effect edge connects an action node with the state node which is the result of the action and is represented by *effect-edge(action(a,t), state(s,t))*. A goal description edge which is represented by *goal-d-edge(state(s,t), goal(g,t))* connects one of goal description states with the goal. A reference edge is represented by *reference-edge(state(uc,t), intent(i,t))* and it connects an intention node with its related user profile state node. An inference edge is represented by *inference-edge(goal(g,t), intent(i,t))* and it connects a goal node with its intention node. A persistence edge is represented by *persistence-edge(state(s,t-1), state(s,t))* and makes it possible to preserve a state which doesn't conflict with the effect of an observed action.

B. Definition of Valid Intention

To resolve intention recognition problem using intention graph, we define some useful concepts.

Definition 1: causal link

*Let a_i and a_j be two observed actions at time steps i and j respectively, where $i < j$. There exists a **causal link** between a_i and a_j, written as $a_i \rightarrow a_j$, if and only if one of the effects of a_i satisfies one of the preconditions of a_j.*

An example is shown in Fig. 4. The effect of observed action a_1 is s_1 and the precondition of observed action a_2 is also s_1. So, there is a causal link between a_1 and a_2. This concept can be extended to goal. In Fig. 4, the effect of a_2 is the goal description of g_2. In this case, we define a causal link between a_2 and g_2 and write $a_2 \rightarrow g_2$

Definition 2: causal link path between action and goal

*Given a intention graph, let a_i be an action observed at time step i and g_j be a goal fully achieved in time step j, where $i < j$. A path that connects a_i to g_j via one or more precondition edge, effect edge, zero or more persistence edge, and a description edge, is called a **causal link path** between a_i and g_j.*

Causal link path is defined between two nodes those are not adjacent. For instance, in Fig. 4, there exists a causal link path between a_1 and g_2.

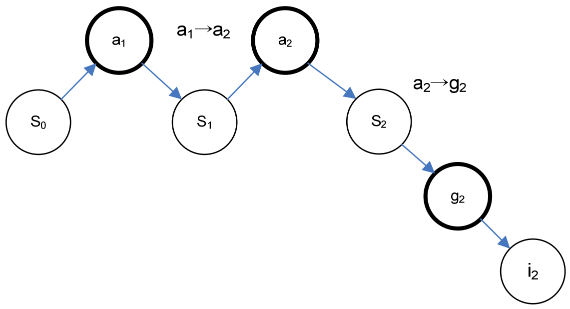

Fig. 4 An Example of Causal Link and Causal Link Path

Definition 3: valid plan

Let g be a goal, and P = <A, O, L>, where A is a set of observed actions, O is a set of temporal ordering constraints, $\{a_i < a_j\}$, over A, and L is a set of causal links, $\{a_i \rightarrow a_j\}$, over A. Let S be the initial states. P is a ***valid plan for g****, given S, if and only if*

1. *the actions in A can be executed in S in any order consistent with O;*
2. *the goal g is fully achieved after the actions in A are executed in S in any order consistent with O.*

An example is shown in Fig. 5. An initial state is S_0, observed action set is $\{a_1, a_2\}$ and goal is achieved after a_1 and a_2 are executed. Then, $P=<\{a_1, a_2\}, \{a_1<a_2\}, \{a_1\rightarrow a_2, a_2\rightarrow g_2\}>$ is a valid plan for g_2.

Definition 4: relevant goal

Given a intention i, a goal g is a ***relevant goal for i*** *if and only if there exists a causal link between g and i, $g\rightarrow i$.*

For instance, in the example shown in Fig. 5, the goal g_2 is the relevant goal of i_2. There exists a causal link between goal g_2 and intention i_2, if and only if a goal g_2 is one of the goal condition of intention i_2.

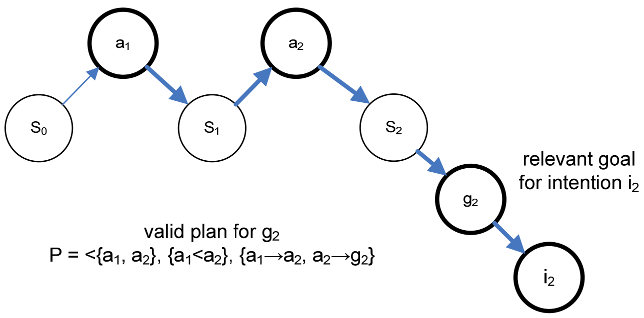

Fig. 5 An Example of Valid Plan and Relevant Goal

Definition 5: valid intention

Let G be a set of relevant goals for intention i, A_o be a set of observed actions, and P_k = < A_k, O_k, L_k> be a valid plan for each g_k in G. Then, i is a ***valid intention*** *if and only if*

1. $A=\bigcup_{k=1,n} A_k$ *where* $n = |G|$
2. $A = A_o$

For instance, in the example shown in Fig. 6, $A_1=\{a_1, a_2\}$, $A_2 = \{a_3, a_4\}$, and $A_1 \cup A_2$ is A_o. So, i_1 is valid intention because the valid plans of its relevant goals cover observed action set.

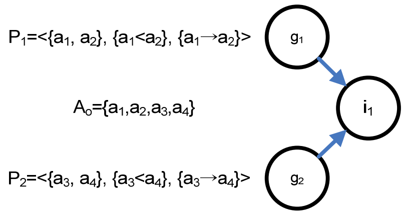

Fig. 6 An Example of Valid Intention

III. Intention Recognition

Intention recognition process based on intention graph has two phases: goal recognition phase and intention recognition phase. In the first phase, plausible goals are recognized through analyzing the observed actions. In the second phase, intentions are recognized based on recognized goals in the previous phase and user profile information.

A. Goal Recognition Phase

In this phase, there are two steps for goal recognition. In the first step, intention graph is extended by adding edges and nodes for observed actions, its effect states, and goals. In the second step, the extended graph is analyzed and goals for actions are recognized.

The graph extending step is as follows. At first, a graph is constructed of initial states. At time *t*, goal extension algorithm shown in Fig. 7, gets the goal descriptions of each instance of goal schema, and converts it to Herbrand instance. That is, the algorithm eliminates every quantifier and converts every variable to instance values. If some of goal description instances are in the current state, it makes a goal node and adds it to the intention graph. A goal node is called fully achieved goal if every goal descriptions are satisfied in current state. Otherwise, we call it partially achieved goal. The algorithm considers fully achieved goals only. So, if goal *g* is fully achieved based on current state, goal node *goal(g, t)* and *goal-d-edge(state(s,t), goal(g,t))* are added. The inputs of goal-extension algorithm are current time, a set of goal schema, and a sub-graph with sate, observed action, edges and a recognized goal set.

```
Goal-Extension (t, G <S, Ao, E, GR>)

For every Gk ∈ G
  For every instance g of Gk
    Get a set of goal descriptions Dg
    Convert Dg to Dg' to eliminate of universal quantifier in Dg
    For every sg in Dg' where sg = not(sg')
      if state(neg(sg'), t)∈S, then
        Add goal(g,t) to GR
        Add description-edge(state(neg(sg'),t), goal(g,t)) to E
    For every sg in Dg' where sg≠not(sg')
      if state(sg,t)∈S, then
        Add goal(g,t) to GR
        Add description-edge(state(sg, t), goal(g,t)) to E
Return with <S, Ao, E, GR>
```

Fig. 7 Goal Extension Algorithm

Open Science Index, Computer and Information Engineering Vol:1, No:1, 2007 waset.org/Publication/11699

Open Science Index, Computer and Information Engineering Vol:1, No:1, 2007 waset.org/Publication/11699

If action *a* is observed at time *t*, the action extension algorithm makes a node *action(a,t)* and adds an edge of *precondition-edge(state(s,t-1), acton(a,t))*. The algorithm adds an effect state node *state(e, t)* for all effects of action *a* and adds *effect-edge(action(a,t), state(e,t))*. If *state(s,t-1)* does not conflict with any *effect(e,t)*, algorithm adds the same state node *state(s,t)* in time *t,* and connects *state(s,t-1)* to *state(s,t)* with *persistence-edge(state(state(s,t-1), state(s,t))*.

```
Action-Extension(t, a_t, A, <S, A_o, E, G_R>)

Add action(a_t, t) to A_o
Get a precondition set P_a and an effect set E_a of a_t from
      action schema set A
Convert P_a and E_a to it's Herbrand base P_a' and E_a'
For every s_p' in P_a' where s_p = not (s_p')
  If state(neg(s_p'), t-1) ∈ S, then
    Add precondition-edge(state(neg(s_p'),t-1), action(a_t, t)) to E
For every  s_p' in P_a' where s_p ≠ not (s_p')
  If state(s_p', t-1) ∈ S, then
    Add preconditoin-edge(state(s_p',t-1), action(a_t, t)) to E
For every s_e' in E_a'
  Add state(s_e', t) to S
  Add effect-edge(action(a_t, state(s_e, t)) to E
For every state(s, t-1) ∈ S
  If state(¬s, t) ∉ S , then
    If state(s, t) ∉ S, then
      Add sate(s, t) to S
      Add persistence-edge(state(s, t-1), state(s, t)) to E
Return with <S, A_o, E, G_R>
```

Fig. 8 Action Extension Algorithm

After last action is processed, the graph is analyzed and proper goals and their valid plans are recognized. A GoalPlan-Recognition algorithm has two parts. At first, redundant goals are pruned. If a goal g_t at time step *t* has no causal link with action at *t*, its goal descriptions are the states from previous time step. If they were not initial states, they actually were results of an action at time *k* where $k < t$. Then there is a goal g_{k+1} which is the same with g_t. The goal g_t is a redundant goal of g_k.

At the second part, the algorithm finds a valid plan following the causal links for each remaining goal. The algorithm returns with GoalPlan list.

```
GoalPlan-Recognition (t, <S, A_o, E, G_R>)

a_t ← the t_th action in A_o
For every g_t ∈ G_R in goal-level t
  If there is not a causal link a_t to g_t ,  then
    Remove g_t from G_R
  else
    A_o' ← {}, A ← {}, CL ← {}
    For every a_k ∈ A_o connected to g_t  by a causal link path
      Add causal link a_k→g_t to CL
      Add a_k to A_o'
      Add a_k to A
    while A ≠ {}
      Remove an action a_l from A
      For every a_k ∈ A_o connected to a_l by a causal link path
        Add a_k→ a_l to CL
        If a_k ∉ A_o' then Add a_k to A_o' and A
    Get all the ordering constraints O over A_o
    Add <g_t, <A_o, O, CL>> to GoalPlan
Return with GoalPlan
```

Fig. 9 Goal and its Plan Recognition Algorithm

B. Intention Recognition Phase

This phase has two steps: intention extension step, graph analysis step. In the first step, intention-extension algorithm gets goal conditions and user profile conditions for each intention schema in schemata set. If all goal conditions are in the recognized goal set and user profile conditions are in current context, then the algorithm adds intention node *intent(i, n+1),* and new state node *state(uc,n+1)*. Also, the algorithm adds reference edge *reference-edge(state(uc, n+1))* to connects user profile state node to intention node, and adds inference edges *inference-edge(goal(gc,k), intent(i,n+1))* to connect every relevant goal node to intention node.

```
Intent-Extension(I, C, <S, A_o, E, G_R, I_R>)

        For every I_k ∈ I
Next:   For every instantce i of I_k
          Get a set of goal-condition GC of i
          Get a set of user-condition UC of i
          For every gc ∈ GC
            If gc ∉ G_R , then
              continue Next
          For every uc ∈ UC
            If uc ∉ CC, then
              continue Next
          Add intent(i, n+1) to I_R
          Add state(uc, n+1) to S
          Add reference-edge (state(uc, n+1), intent(i, n+1))
          Add inference-edge (goal(gc, k), intent(i,n+1))
Return with <S, A_o, E, G_R, I_R>
```

Fig. 10 Intention Extension Algorithm

In the second step, the intent-recognition algorithm gets a set of relevant goals for each intention in an intention schema set. A_I' is a union set of all actions in valid plans of relevant goals. If A_I' is same with observed action set A_o, *I* is the valid intention.

```
Intent-Recognition(<S, A_o, E, G_R, I_R>)

ValidIntention ← {}
For every I ∈ I_R
  Get a relevant goal set G for I
  A_I' ← {}
  For every g in G
    Get a valid plan set P=<A, O, L> for g
    while A ≠ {}
      Remove a_k from A
        If a_k ∉ A_I' then Add a_k to A_I'
  If A_I' = A_o then
    Add I to ValidIntention
Return ValidIntention
```

Fig. 11 Intention Recognition Algorithm

The algorithm returns with the valid intention lists.

C. Algorithm Complexity

Our algorithms have polynomial size and time complexity. The first 3 algorithms are based on Jun Hong's Goal Graph algorithm, and it is proved polynomial size and time in [5]. Therefore we prove intention recognition phase algorithm in this section.

Theorem 1: (polynomial time and space)

Consider an intention recognition problem with l_a observed actions, a finite number of object instance at each time step. Let n be the number of object instance, l_i be the number of intentions in intention schema set, l_g be the number of goals in goal schema set, m_r be the maximum number of relevant goals of an intention, m_u be the maximum number of user condition of an intention, and m_g be the maximum number of goal condition. Then, the space size of the intention graph and time needed to recognize all valid intention are polynomial in l_a, l_i, m_r, m_u, m_g, and n.

Proof.

The maximum number of intention nodes is $l_i \cdot n$, because there can be no same intention node in the intention graph generated by intention-extension algorithm. The number of user condition node is $m_u \cdot l_i$, and the number of edges is $(m_g+m_u) \cdot l_i$. Since the intention recognition algorithm adds no nodes and edges, the space size of our algorithm is $O((1+m_g+2m_u) \cdot l_i)$.

The time complexity is $O((m_g+m_u) \cdot l_i \cdot n)$. ■

IV. Briefcase Domain

We apply Intention Graph to briefcase domain [7]. It is modified to include intention and user profile information. The modified problem is shown in table 1. Physical objects packing in the briefcase can be transferred between three places. User profile can be any kind of information in any representation. As user profile representation is not our issues, we use user's occupations in text style.

There are four kinds of action schema, goal schema, and intention schema. Schema examples are shown in Fig. 12. Action and goal can have parameters. An action schema has preconditions and effects. A goal schema contains desired states. An intent schema has its goal condition and user condition.

```
(:action move
        :parameters ((object ?b) (place ?l ?m))
        :precondition (and (briefcase ?b) (at ?b ?l) (not (= ?m ?l)) )        (a)
        :effect (and (at ?b ?m)
                (not (at ?b ?l))
                (forall (?x)
                    (when (and (object ?x)  (in ?x ?b))
                        (and (at ?x ?m) (not (at ?x ?l)))))) )
(:goal keep-object-at
        :parameters ((object ?x) (place ?l))                                  (b)
        :description (and (at ?x ?l) (not (in ?x))) )
(:intent studying
        :goal-condition (keep-object-at dictionary office)
        :user-condition (usr_occupation(student)) )                           (c)
```

Fig. 12 Examples of Schema

TABLE I
Briefcase Domain Explanation

Classification	Value
Physical object	a briefcase, a dictionary, a checkbook, a pencil
Places	home, office, shop
Action Schemata	. Moving the briefcase from one location to another . Putting a physical object in the briefcase . Taking out a physical object from the briefcase . Printing a check
Goal Schemata	. Keeping a physical object at a location . Moving a physical object from one location to another . Printing a check for a person . Walking into a location
Intention Schemata	. He/She would like to come home from work . He/She would like to write a story . He/She would like to study English . He/She would like to pay for something
User Profile	. user_occupy

In this domain, a physical object *briefcase* is instantiated as B, a place *home* as H, a place *office* as O, and a *dictionary* as D. The initial states are given as {at B H, at D H} and actions are observed in the sequence of {put_in D H < move B H O < take_out D}. After graph construction step finish in goal recognition phase, the intention graph has 9 goal nodes. During the goal pruning step, 6 goals are removed. With the three goals and its valid plans, our algorithms find valid intentions during intention recognition phase. The results graph is shown in Fig 14.

Open Science Index, Computer and Information Engineering Vol:1, No:1, 2007 waset.org/Publication/11699

Open Science Index, Computer and Information Engineering Vol:1, No:1, 2007 waset.org/Publication/11699

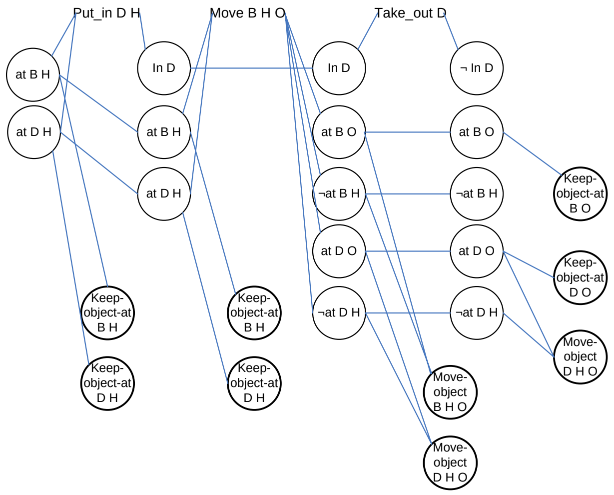

Fig. 13 An Intention Graph after Graph Construction Step

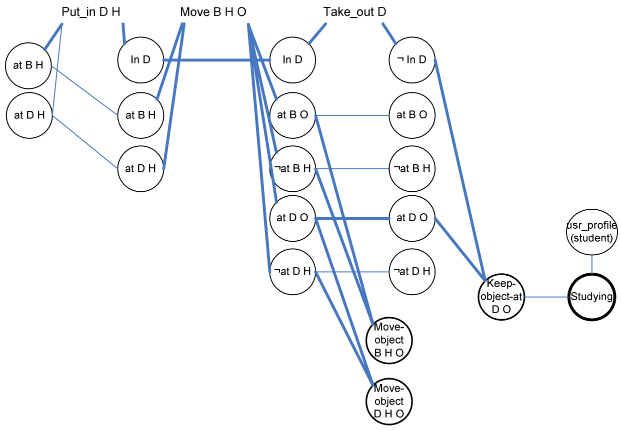

Fig. 14 An Intention Graph after Intention is Recognized

V. Conclusion

We have discussed intention recognition problem and have proposed an approach to recognize valid intentions using intention graph. It is inspired by the idea of Goal Graph and GraphPlan. The Intention Graph is extended by action nodes and its effect sate nodes. After observing actions is finished, the graph is analyzed. And then, valid intentions are recognized based on relevant goals and user profile information under current context. The algorithm has polynomial time and space complexity.

Although the work reported here is encouraging, much remains to be done before it can be considered complete. The most obvious defect of the prior model is that it considers closed world assumption. Some information could be missed or vague, but we can't handle it. Another weakness is that just one action can be observed at a time step in the Intention Graph. Two or more actions can be happen in the real world, especially HRI domain. Work is currently under way to address these issues.

References

[1] C. Breazeal, "Social interactions in HRI: The robot view", *IEEE Trans. Systems, Man, and Cybernetics*, vol.52, no.6, pp 181-186, May 2004.

[2] Farrah Wong, Kwang-Hyun Park, Dae-Jin Kim, Jin-Woo Jung and Zeungnam Bien, "Intention reading towards engineering applications for the elderly and people with disabilities", in *International Journal of ARM*, vol. 7, no. 3, pp. 3–15, September 2006.

[3] Ann Elwan, *Poverty and disability – a survey of the literature, social protection discussion paper series*, Social Protection Unit, Human Development Network, The World Back, no 9932, pp. 5, Dec. 1999.

[4] M. Rahimi and W. Karwowski (Eds.), *Human-Robot Interaction*, London: Taylor and Francis, pp.2, 1992.

[5] Jun Hong, "Goal recognition through goal graph analysis", *Journal of Artificial Intelligence Research*, vol.15, pp. 1-30, 2001.

[6] Avrim L. Blum, Merrick L. Furst, "Fast planning through planning graph analysis", *Artificial Intelligence*, pp. 281-300, 1997.

[7] E. P. D. Pednault, "Synthesizing plans that contain actions with context-dependent effects", *Computational Intelligence*, vol. 4, pp. 356-372.

[8] Chiung-Hon Leon Lee, and Alan Liu, "An intention-aware interface for services access enhancement", in *Proc. of the IEEE Int. Conf. on Sensor Networks, Ubiquitous, and Trustworthy Computing*, vol.2, pp.52-57, 2006.

[9] Tomomasa Sato, Yoshifumi Nishida, Junri Ichikawa, Yotaro Hatamura, and Hiroshi Mizoguchi, "Active understanding of human intention by a robot through monitoring of human behavior", *Proc. of IROS '94,* vol.1, pp.405-414, 1994.

[10] Yoshinori Kuno, Nobutaka Shimada, and Yoshiaki Shirai, "Look where you're going", *IEEE Robotics & Automation Magazine*, pp.26-34, March 2003.

[11] D. J. Kim, W. K. Song, J. S. Han, Z. Zenn Bien, "Soft computing based intention reading techniques as a means of human-robot interaction for human centered system", *Soft Computing*, pp. 160-166, 7, 2003.

[12] Toru Yamaguchi,Shinya Mizuno, Takuya Yoshida, and Tomomi Hashimoto, "Cooperative works for agent robot and human using robot vision based on the model of knowledge, emotion and intention", *Proc. of IEEE SMC '99*, vol.2, pp.987-992, 1999.

[13] Soshi Iba, Christiaan J. J. Paredis, Pradeep K. Khosla, "Interactive multi-modal robot programming", In Proc. of the 9th Intl. Symposium of Experimental Robotics, June, 2004.

[14] Z.Zenn Bien, Kwang-Hyun Park, Jin-Woo Jung, and Jun-Hyeong Do, "Intention reading is essential in human-friendly interfaces for the elderly and the handicapped", *IEEE Trans. On Industrial Electronics*, vol.52, no.6, December 2005.

So-Jeong Youn received the B.S and the M.S degree in engineering from the Department of Computer Science of the Sogang University, Seoul, Korea in 1991 and 1993, respectively.

She was a researching member of Electronics and Telecommunications Research Institute (ETRI), Daejon, Korea, from 1993 to 1997. After she completed the course of a doctorate in computer science, she became a faculty of Chungkang college of Cultural Industries. She was a professor of Department of Computer Networks from 2000 to 2006. Now, she is studying for her dissertation about intention modeling and recognition in Sogang University.

Kyung-Whan Oh received the B.S degree in Mathmatics from Sogang University, Seoul, Korea in 1978, and the M.S. and Ph.D. degrees in Computer Science from Florida State University, FL. USA, in 1985 and 1988, respectively.

He is currently with the department of Computer Science at Sogang University, Seoul, Korea, where he is a Professor.