JANITORON-THE REFRIGERATOR INCHARGE

SWATI MEHTA*; NIDHI ARORA**

*Visiting Faculty,
Innovative School of Business Management,
Ahmedabad, Gujarat, India.
**Research Engineer,
Solusoft Technologies Pvt. Ltd.,
Ahmedabad, Gujarat, India.

ABSTRACT

Intelligent agent systems are becoming popular and more important in our day-to-day activities and users are expecting more intelligent services from them. The paper proposes a third level of sophistication agent capable of flexible autonomous behavior. It provides volunteer information or services to a user, without being asked, whenever it is deemed to be appropriate. A prototypical agent-based refrigerator system has been described, where the environment is challenging; more specifically, it is dynamic, unpredictable and unreliable. The refrigerator agent-‘Janitoron’ with high complexity is expected to have all the characteristics in the ideal situation ranging from autonomy, informativity and both reactivity and pro-activity. The proposed agent maintains a balance between reactivity and pro-activity. An agent senses changes in its environment and responds to these changes. On one hand, the agent’s actions are influenced by environmental changes and on the other hand, the agent’s plans and actions are influenced by its goals. Being proactive and reactive, Janitoron behaves more like humans in the way it deals with different situations.

The paper describes the basic working of the refrigerator agent. The agent receives a percept containing information about the food kept at certain location in the refrigerator. Since the agent has no existing knowledge about the food, this percept results in updating of the agent’s beliefs with new knowledge. This in turn generates a goal to take actions, which then leads to a plan being selected and executed. This way the agent can monitor status of food kept in refrigerator if the user don’t have time do so.

KEYWORDS: Agent Specifications, Dynamic Environment, Intelligent agent, Percept Sequence, Refrigerator Agent.

1. INTRODUCTION

Though agents are a rapidly growing area of research in the artificial intelligence, most past definitions of agent are behavioral—agents are programs that exhibit certain hard to specify behaviors. In past decades, software developers created massive software programs that often performed a wide variety of tasks. At the same time however, there has been a shift from the development of massive programs containing millions of lines of code to smaller pieces of code.
Each module performs a well defined, focused task rather than thousands of different tasks as opposed to the legacy systems. Software agents are the latest innovation in this trend towards splitting complex software systems into components. An agent is a body of software that provides one or more useful services, includes the ability to act autonomously without requiring explicit direction from a human being, has the ability to determine what actions to take in given conditions and includes the ability to interact with other agents including human beings.

2. BACKGROUND

Years ago technology of software agents has been recognized as a fast emerging area of research [Nwana, 1996], [Jennings, 1996] and one of the rapidly growing areas of information technology. Agents are described as problem solving entity [Wooldridge and Jennings, 1995] which exhibit properties like Autonomy, Social ability, Proactiveness and Responsiveness. Reactive agents give fast response to every change in their environment. They do not waste time with the long reasoning processes. Nevertheless, they can be able to perform complex tasks. Yoav Shoham [Shoham, 1993] presented a computational framework which describe the state of an agent consists of components such as beliefs, decisions, capabilities, and obligations. An approach that exploits situated automata for reactive agent architecture is described in [Kaelbling, 1991]. Petrie Maes [Maes, 1991] developed a reactive agent architecture that is composed of modules organized into network. The idea of believable agents [Bates J., 1994] put forward include the appearance of reactivity goals emotions and situated social competence among others. [Jenning et.al., 1996], described Advanced Decision Environment for Process Tasks for developing an agent-based infrastructure for managing business processes. Multi-agent Systems approach and agent-oriented programming are suitable for the open system programming [Burkhard, 1994] which has the following features: continuous availability, extensibility, decentralized control, asynchrony, inconsistent information, arms-length relationships. The use of Internet-base agents and their compatibility with the World-Wide Web has been studied by [Petrie, 1996].

A soft computing framework for intelligent agents [Seung-Ik Lee, 2001] was proposed to overcome the shortcomings of the conventional frameworks. The framework consists of two parts: construction and analysis part. In the construction part, the combination of rule-based systems and evolutionary algorithms is used to construct an intelligent agent. In the analysis part, evolutionary activity statistics, schema analysis, and observational emergence are applied to the analysis of the evolved agent in both evolutionary and behavioral perspectives. Architecture of an intelligent agent for speech recognition and translation [Abd Manan, 2006] was developed. Agent-based speech recognition and translation is a prototype that integrated speech process and translation process in one system. This prototype is able to get an input from user’s voice in English and send back the result in text. All types of agents work on rules which are triggered and lead to specific plans which is executed to perform actions [Winikoff et.al., 2001]. A plan is a way of realizing a goal which is basically executed through the effecter.

3. INTELLIGENT AGENTS

An agent is a computer system situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives. Autonomy is difficult to achieve
precisely in reality, but the system should be able to act without the direct intervention of humans (or other agents), and should have control over its own actions and internal state. It not only assists people in doing their work but it can also act on their behalf. Intelligent agents work by allowing people to delegate work that they could have done, to the agent software. Agents can perform repetitive tasks, remember things you forgot, intelligently summarize complex data, learn from you and even make recommendations to people. Agent should not simply act in response to the environment; it should be able to exhibit opportunistic, goal-directed behavior and take the initiative wherever appropriate.

The agent is expected to be autonomous and must be able to sense the changes in the surrounding environment. The refrigerator agent must be independent in taking its decisions. The agent will be proved to be useful as the environment inside the refrigerator is challenging. More specifically, the environment is dynamic, unpredictable and unreliable. The environment is dynamic in it cannot be assumed to have same food in all the racks. The environment is unpredictable in that it is not possible to predict which food item is going to get spoiled within next few hours. Finally, the environment is unreliable in that the actions that an agent can perform may fail for reasons that are beyond an agent’s control.

FIG. 1- INTELLIGENT AGENT INTERACTING WITH ENVIRONMENT
4. REFRIGERATOR INTELLIGENT AGENT

A refrigerator intelligent agent works exactly like a typical agent and is situated in dynamically changing environment. Since the agent’s environment is dynamic, the agent must respond to significant changes in its environment. The agent can act in both reactive and proactive manner. Sometimes the agent is reactive, responding to changes in the refrigerator. Some other times the agent has to pursue goals over time, means, it is proactive. The agent continues to attempt to achieve a goal despite failed attempts.

The refrigerator agent possesses some other properties like:

- **Mobility**: the ability to move around in the refrigerator from one rack to the other
- **Veracity**: the agent does not knowingly communicate false information by blowing wrong lights.
- **Benevolence**: the agent does not have conflicting goals and the agent therefore always try to do what is asked of it.
- **Rationality**: an agent acts in order to achieve its goals insofar as its beliefs permit.
- **Learning/adaptation**: the agent tries to improve performance over time.

5. WORKING OF JANITORON

A refrigerator agent receives a percept containing information about the foul smelling food at a certain location at certain rack. Since the agent has no existing knowledge about the food, this percept results in updating of the agent’s beliefs with knowledge of this new one. This in turn generates a goal to alarm the user by blowing appropriate light, which then leads to a plan being selected and executed.

5.1 INITIAL SETTINGS

To develop the agent, we begin with the interface to the agent. Before the agent can be put into refrigerator for use, some initial settings are required. The following settings are to be considered:

- **Time period settings**
  - Cycle time $t_1$ for cleaning the refrigerator
  - Time after which the user is to be warned for food kept for more than $t_2$ hours.
- **Distance between two racks in the refrigerator**
- **Total number of racks in the refrigerator**
• The area per rack in the refrigerator

• The ignorance time t3 for which the agent has to wait after turning on the warning lights.

5.2 AGENT RULES: CONDITION-ACTION PAIRS

The agent searches for an appropriate action, given a percept sequence. For this, the agent has to first formulate a goal. Then it searches for a sequence of actions that would be used to solve the problem. The actions are then executed to change the current state and again the cycle is repeated until the goal is achieved. The following generalize function shows the refrigerator agent’s behavior:

While interacting with the environment, the agent has to choose appropriate actions for each percept. The agent act according to states, and which actions they take are determined by rules of the form “When P is true of the state of the environment, then the agent should take action A.” The agent may use a rule of the form “If the food kept at a rack R has chances of getting stale, glow the warning light.”

5.3 PERCEPTS AND ACTIONS

A percept is an item of information received from the environment by some sensor. Here, the refrigerator agent may receive information such as the location of a foul smelling food and its intensity. An agent may also obtain information about the environment through sensing actions. Identification of percepts to be obtained is driven by the information the agent system requires in order to achieve the goals identified in the initial settings. Many percepts result in update of the agent’s knowledge as well as potential action on the part of the agent. Knowledge updates resulting from the percept should be explicitly identified.

Percepts often require some processing in order to extract the information that is of value to the agent system. The agent would warn the user for the following:

• Food having chances of getting stale/rotten

• Food that is already rotten

• Food kept for more than t2 hours.

• Refrigerator needs cleaning

• Spilling of food

• Foul smelling food

The percepts of the proposed agent have the following form:

• Name – No problem at current rack.
The actions of the agent are as follows:

- **Name** – Action in response to percept 'No problem at current rack'.
- **Description** – Indicates that the agent has to move to another rack and is expected to ensure cleanliness at the intended rack.
- **Parameters** – To jump to another rack, the agent must know the distance between two racks and the area which the agent has to perceive on the rack.
- **Temporality** – Durational/instantaneous. Since it is durational, the length of time is to be indicated for which the cycle is to repeat.
- **Failure/fault detection** – The appropriate warning light is to be blown whenever some fault is noticed.
- **Partial change** – If the warning light goes unattended, the agent has to start doing the cleaning work on the behalf of user.
- **Side effects** – If the cleaning is done by the agent, it may result in performance reduction of agent.

### 5.4 AGENT SPECIFICATIONS AND FUNCTIONS

double linearVelocity;
double angularVelocity;
double cleanDuration;
double staleFoodDuration;
double ignoranceDuration;
Clock JanitoronClock;
bool update_setting()
{
    JanitoronClock.Reset();
SetSpeed(linearVelocity, angularVelocity);

bool status = !JanitoronClock.Expired(Time(cleanDuration));

return status;

}  

bool turn_stale_food_indicator_on()
{
    bool status = JanitorClock.Expired(Time(staleFoodDuration));
    JanitorClock.reset();
    return status;
}

bool turn_off_indicators()
{
    bool status = JanitoronClock.Expired(Time(ignoranceDuration));
    return status;
}

6. CHARACTERISTICS OF TASK ENVIRONMENT

6.1 PERFORMANCE MEASURE

The performance of the agent can be measured using the following criterion:

- How actively the agent moves from one location to another in the refrigerator i.e. its speed of movement.

- How precisely the appropriate warning lights are turned on and off i.e. the ability of the agent to choose the right rule at the right time.
6.2 ENVIRONMENT

As the agent is operating inside the refrigerator, the agent’s environment consists of:

- Area of one rack
- No of racks in the refrigerator
- Starting location from left corner of lowest rack
- Food vessels/packets/vegetables/bottles

6.3 ACTUATORS AND SENSORS

The agent has a set of actions that it can physically perform. The actions performed by the agent are capable of changing its current state. The agent has an associated action policy that states the conditions under which the agent may, may not, or must do some actions. The actions that the agent can take, as well as its action policy, are clearly stated in terms of rules, before it is put into practice.

The agent performs actions based on the matched rules. For that it has to continuously move from one location to another and his movements can be:

- Move forward
- Move backward
- Turn left by 30, 45 and 90 degree
- Turn right by 30, 45 and 90 degree

Some specialized sensors such electronic snot can be integrated into Janitoron with for identifying the different types of smells for the food. It is also used to pick up the smells produced by bacteria and pathogens as they break down foods, to guide it to take appropriate actions. The artificial mucus layer in electronic snot aids the agent more adept at detecting and distinguishing food odors.

Depending on the percept obtained from sensors, the agent blows the warning lights as follows:

- If some food is kept unused/ unnoticed for more than \( t_2 \) hours – orange light
- If something in the refrigerator has spill or if the refrigerator needs cleaning – green light
- If some food is already rotten or has chances of getting rotten or some food is smelling foul – red light
If any light is ignored for more than $t_3$ hours, the agent will clean the rack itself thereby blowing the Blue light.

The refrigerator is equipped with the listed lights along with a Thanks Button. The user has to press the button for putting off the warning lights.

7. CONCLUSION

Intelligent agent has been around for years, but the actual implementation is still in its early stages. With their effectiveness, these agents are attracting considerable interest. The concept and technology behind intelligent agents are making their transition from researcher’s ideas to their implementation and now to the applicability in business and at home. As agents gain a wider acceptance and become more sophisticated, they will soon become a major part in our life. Intelligent agents in reality cannot completely replace human beings, but they can act as catalyst in our routine business and/or household activities.

8. REFERENCES


