Conceptual Flow-Based Modeling of Agents

Sabah Al-Fedaghi
Kuwait University

Abstract

Agent-based systems have emerged as an important next-generation automation paradigm. One of the aims of agent-oriented research is to simplify agent design by introducing a conceptual abstraction layer for interacting agents. This paper provides a new multilevel conceptual foundation for modeling of agents, their relationships, and their interactions based on BDI concepts. The methodology utilizes a flow model of five stages. It incorporates all BDI aspects by assigning agents to information/beliefs, goals, plans, and actions spheres.

1. Introduction

Current methodologies for analyzing requirements for the design of agents rely on typical tools such as flow charts, UML, and Petri nets. These methods seem to lack compatibility with an agent-oriented, problem-solving technique that combines external and internal processes. Consider Figure 1, used in an agent standardization document [6].

Several disturbing aspects can be observed in Figure 1. The agent is modeled as an oval shape, lacking internal structure. Goals and intent are supposed to be interior features of the agent. "Communication" is implied to be partially inside that agent. The observer is unsure about the origin of the message. The description of the origin in this regard is analogous to describing spoken words as originating from the body of a person instead of from his/her mouth. Additionally, the significance of two types of arrows in Figure 1 is not clear. It seems that the solid arrowheads represent a flow of information. If this is the case, then how this flow reaches and leaves such entities as "Convert to transport form" is unclear. This element seems to be a process, so the arrows indicate "data control" as in a typical flowchart. Data flow occurs according to availability, while control flow specifies order of execution that depends on availability – e.g., wait until the arrival of another input. The location of the "Convert to transport form" process is not clear, but apparently it is either in the communication channel or inside the agent. The heterogeneity of the elements (boxes) involved is evident. The dotted line indicates some relationship between goals, intent, acts, messages, and the flow of data or control.

A better conceptual picture can be achieved with a simple description of the generic components involved in this message delivery/transportation service. We will introduce this high-level description utilizing a flow model. The flow model is a uniform method for representing things that "flow," i.e., are exchanged, processed, created, transferred, and communicated. "Things that flow" include information, materials (e.g., manufacturing), and money (finance), as well as goals (e.g., create goals, specify goals for others), plans (receive plans, disclose plans), and actions(create actions, receive actions).

The concept of flow is widely used in many fields of study. In economics, the circular-flow model of goods is well known, and in management science there is the supply chain flow. In computer science, the classical model of flow is the 1949 Shannon-Weaver communication model representing electrical signal transfer from sender to receiver.

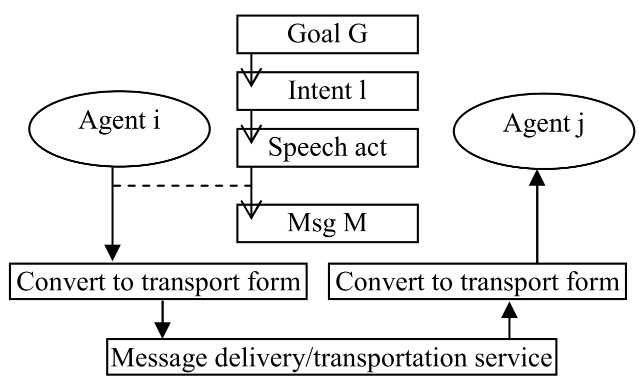

Figure 1. Message between agents (from [6])

978-0-7695-3571-5/09 $25.00 © 2009 IEEE
DOI 10.1109/GCIS.2009.30

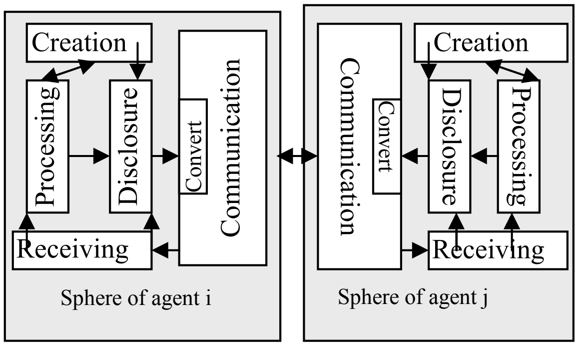

Figure 2. Flow description of information interaction between two agents.

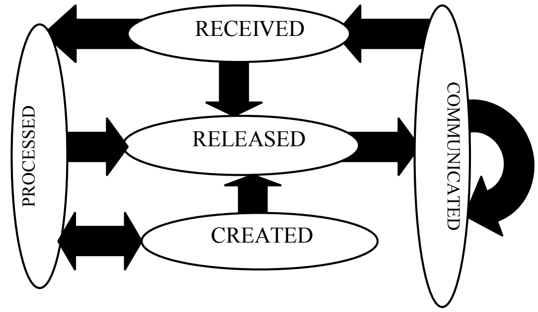

Figure 3. Transition states.

Figure 2 reflects the concept of "flow" in terms of states: information being transmitted, information in the channel, information being received, and information feedback. Flow of information means movement from one information sphere (the sender) to another information sphere.

In the flow model (FM) approach, the concept of flow is understood in a more comprehensive way. FMs have a number of desirable features, including spatial arrangement of their components relative to each other and to time as well as representation of links between the components indicating the flow of items. To simplify the review of FMs, we introduce flow in the next section in terms of *information* flow.

Considering the flow model description of agent interactions shown in Figure 1, the informational spheres of the agent are shown in Figure 2. Only five generic stages are applied to each agent. The arrows strictly represent the flow of messages. The other spheres, such as goals, plans, and actions, have the same uniform representation.

2. The flow model

The flow model (FM) has been used in several applications such as communication [1] and human-machine interaction [2]. This section reviews the basic model and includes new illustrations of its general features.

In an FM, information goes through a sequence of states as it moves through stages of its life cycle:

1. Information is received (i.e., it arrives at a new sphere, comparable to passengers arriving at an airport).
2. Information is processed (i.e., it is subjected to some type of process, e.g., compressed, translated, mined).
3. Information is disclosed/released (i.e., it is designated as released information, ready to move outside the current sphere, like passengers ready to depart from an airport).
4. Information is transferred (disclosed) to another sphere (e.g., from a customer's sphere to a retailer's sphere).
5. Information is created (i.e., it is generated as a new piece of information using methods such as data mining).
6. Information is used (i.e., it is utilized in some action, analogous to police rushing to a criminal's hideout after receiving an informant's tip). Using information indicates exiting the information flow to enter another type of flow such as actions. We call this point a gateway in the flow.
7. Information is stored; thus, it remains in a stable state without change until it is brought back into the stream of flow again.
8. Information is destroyed.

The first five states of information form the main stages of the stream of flow, as illustrated in Figure 3.

When information is stored, it is in sub-states because it exists in different stages: as created (stored created information), processed (stored processed information), and received (stored received/raw information) information. The five-stage scheme can be applied to humans and organizations. It is reusable because a copy of it is assigned to each agent.

The five information states are the only possible "existence" patterns in the stream of information. To follow the information as it moves along different paths, we can start at any point in the stream. Suppose that information enters the processing stage, where it is subjected to some process. The following are ultimate possibilities: (a) It is stored. (b) It is destroyed. (c) It is disclosed and transferred to another sphere. (d) It is processed in such a way that it generates implied information. (e) It is processed in such a way that it generates new information (e.g., comparing certain statistics generates the information that *Smith is a risk*). (f) It is used to generate some action (e.g., upon decoding or processing the information, the FBI sends its agents to arrest the spy who wrote the encoded message).

The storage and uses/actions (gateway) sub-stages can be found in any of the five stages. However, in the release and transfer stages, information is usually not subject to these sub-stages, so we apply these sub-stages only to the receiving, processing, and creation stages without loss of generality. Implicit in the model is the fact that information may be duplicated through copying. It can also be destroyed at any point in the model.

To illustrate the actions (gateway) sub-stage, consider a person in Barcelona who uses the Internet to ask a person in New York whether it is raining in New York. The New Yorker's receipt of the query triggers an action, such as physical movement to look outside to check whether it is currently raining. The gateway in his/her information system transforms the information flow to an action flow. After observing whether it is raining, he/she triggers another information flow by creating a response in his/her creation stage. The response flows though links and crosses from one information sphere into another.

3. Modeling BDI agents

Chang-Hyun and Einhorn ([7] and [8]) introduced what they call Agent-based Modeling Technique and Agent-based Software Development Process as "systematic and realistic processes" to construct BDI agent-based software. A BDI agent is a software agent with mental attitudes: Beliefs (informational), Desires (motivational), and Intentions (deliberative). They used "external" use cases as general plans for specific services and then refined these plans into goals using internal use cases. "The beliefs are determined for each goal by analyzing each goal's plans and determining what beliefs will be necessary for its completion" [7]. In this section, we partially follow their methodology, which conceptualizes agents and their relationships in specific case studies of creation and bookkeeping of weather notices. We then remodel their description utilizing FM representation. This makes it easier to demonstrate the descriptive power of FM.

Chang-Hyun and Einhorn's [7] case study aims at developing a tool to aid weather forecasters in providing notices to state districts. The system provides weather notices for a variety of events (frost, severe weather, freezing rain). Customers use these notices as a warning that they may need to take action in response to an event. The customers usually want the notices delivered to each district where the notices are valid. The system functions are specified through external use cases in several iterative steps. The following describes a sample External Use Case.

Name: CreateNotice: A forecaster (a) identifies the need to submit a notice or notices in a region, (b) starts the notice interface, (c) selects the state to submit notices to, (d) selects the districts to submit notices to, and (e) creates and submits the notice to the system. The system recognizes and formats the notice, and then delivers the notice.

According to Chang-Hyun and Einhorn [7], conceptual agents are candidate agents to which BDIs are later assigned. Linguistic analyses (e.g., nouns) are used to differentiate candidate agents from problem statements and external use cases. The following conceptual agents of the Notice Management System are identified: Notice, Notifier, System, Forecaster, Customer, Weather, District, Web Page, Fax, and Delivery. Figure 4 shows Chang-Hyun and Einhorn's Conceptual Agent Relation Diagram for Service: createNotice.Here. The oval is used to represent external agents, and rectangles to represent internal agents. The semantics of the arrow are not clear. It is certainly "control flow," since the arrows indicate an order of operations. They may also indicate a flow, as in the case of delivering notes. The figure contains heterogeneous entities, including actors (e.g., forecasters), entities subject to action such as notes (notices), and processes (delivery).

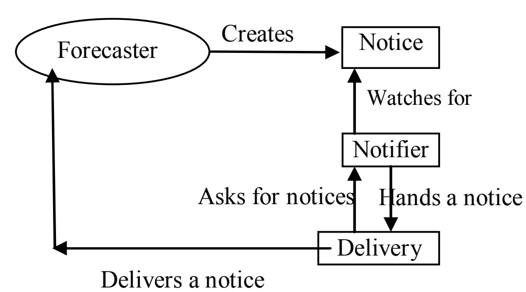

Figure 4. Agent Relation Diagram for Service: createNotice (From [7]).

The FM presents a better design, one in which spheres represent a collection of stages of information flow with recognized boundaries and logical structure that specifies inside/outside components. It includes four infospheres for modeling: forecaster, system, notifier, and users with notes, and the flow of requests among them. Figure 5 shows the FM description.

We assume that forecasters are triggered to create notes by some event from an outside sphere (the dotted arrow at point 1 in figure 5). Let us assume that the forecaster receives information about weather conditions from a national center through communication and receiving stages. The forecaster then processes this information to create a note that flows to the system sphere (point 3). It is transmitted through the communication stages to arrive at the receiving stage of the system, where it is stored.

Chang-Hyun and Einhorn's [7] description of this case study is ambiguous with regard to the issue of the notifier creating a request. We assume that the notifier is an agent (subsystem) that handles communication with the user. Thus, at point 1, the user creates a request for a note that flows to the system and then to the notifier.

The request then moves to the system sphere through different stages to trigger the retrieval (point 7) of the appropriate note that flows back (point 5) to the user.

Chang-Hyun and Einhorn [7] then decompose the system's functions from external use cases by building internal use cases into the goals (desires). A system service described in an external use case is decomposed into one or more goals. Each decomposition step creates an internal use case. An internal use case describes the plans for each identified goal to provide a system service described in an external use case. Nevertheless, the approach needs to clearly distinguish between the spheres of information, goals, plans, and actions.

In their approach these are mixed in a mesh of descriptions that is difficult to analyze. Alternatively, we will utilize our FM methodology to model agents' goals, plans, and actions. In FM, goals, plans, and actions are treated uniformly as information, as described below.

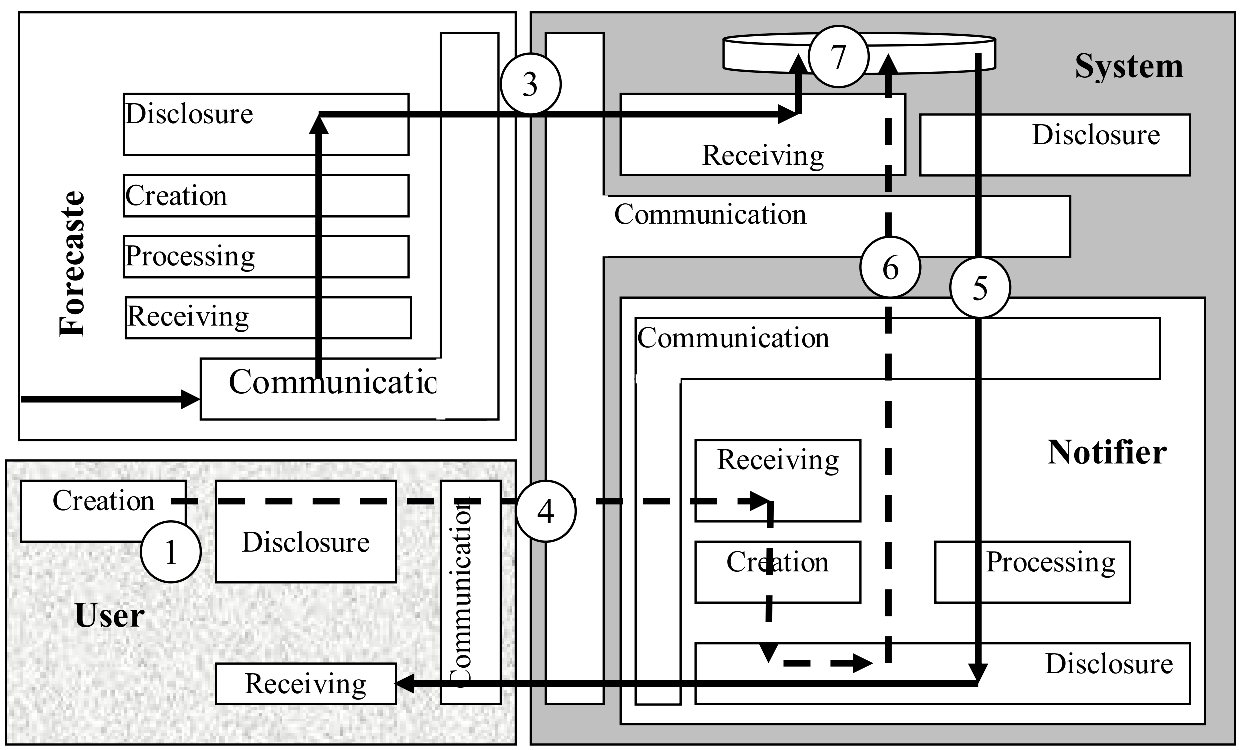

Figure 5. FM representation corresponding to Figure 8.

4. BDI spheres

Belief-Desire-Intention (BDI) architecture [3] is a well-known theoretical foundation for describing rational agents. In BDI, beliefs refer to *informational* attitudes, desires to *motivational* attitudes, and intentions to deliberative attitudes of agents. For software agents, these three notions are represented as beliefs (e.g., objects), goals (e.g. states to be achieved), and plans (procedural recipes) [Rao and Georgeff 1995]. Exterior and interior events create messages that activate goals, which trigger plans. Beliefs influence and are influenced by this process and may cause internal events.

In FM, beliefs, goals, plans, and actions are elements of different spheres. Each, as information, can be represented by the five stages scheme, with flow among them. In the goals sphere, goals are received (from another agent), created (internally), processed (deliberated), disclosed, and communicated in a similar fashion to information. Plans and actions can also be in any of these states. Spheres can contain sub-spheres from any type.

Such an approach is in line with the agent/human metaphor such that a unified model can be used for the description of the characteristics of agents and of the human mind. Both have a structure comprising several interrelated flow stages. For example, in the "information processing" field, a human mind is viewed as a symbol-manipulating machine that has some features similar to those of agents, such as searching, comparing, and storing. The Shannon-Weaver model and its extensions are usually the starting point for this "computer metasphere." FM can also be used to apply human mind features such as desires and goal-setting to the structure of agents.

In pursuing such a venture, we need to apply the notion of flow to non-informational entities (e.g., desires). According to Devlin [5], "One of the most important (and fundamental) cognitive abilities possessed by man is the ability to *individualate* 'objects', that is to say, to see objects as *objects*…It should be noted that individuation of individuals does not at all require them to be atomic entities, incapable of subdivision within the scheme of individuation." In our case, we need to tie our flow model to non-informational phenomena such as goals, plans, and actions. For example, goals can be visualized as phenomena that refer to future states, thus they are different from information. Since it is possible to identify "goals," the ontology and nature of a "goal" is not of concern here. This is similar to what we have been calling "information"; as long as we can recognize a piece of information, we can recognize its flow in the scheme of individuation. Also, we are not concerned with the structure of a piece of information, a piece of goal, a piece of plan, or a piece of action. A piece of information can be a single digit or a whole file; the only requirement is to have the feature of "individuation" as a discrete element of flow with recognized starting and ending points.

Thus, in our case study, we have four types of flow: information flow, goals flow, plans flow, and actions flow. For example, pieces of action are discrete physical units defined in terms of their corresponding pieces of information that express the action. A piece of action, say ρ, is well defined in terms of its corresponding pieces of information. If ρ is the physical act described in "The robot turns on a switch," then ρ is the phenomenon of "action" that corresponds to the information expressed by the linguistic construct *The robot turns on the switch*. This is a reversal of what is called the "correspondence theory" that maps language to reality. Action ρ and its flow are certainly observable: "The robot A turns on a switch" causes "The robot B turns on a switch," etc. Marketing is based on triggering (flow to) the act of "buying." A system of gears is a flow of actions (movement) from one gear to another. Actions are created, received, processed, disclosed, and communicated just as in any flow system.

We propose to apply the flow model to pieces of goals, pieces of plans, and pieces of actions. We can use different terminologies to express the same phenomena. Instead of "receiving goals" we may use the term *planting* (of desire) to name the stage of "importing" a desire to achieve a future state. We may also use *retention* instead of storage. Thus, the goals sphere is a world where the "things that flow" (currency) are goals. If we want to know what goal it is that has just been generated or created, we look at its corresponding linguistic expression. However, we always keep in mind that goals are not information. In the human mind, desires are certainly different from information (knowledge). If information arrives at the infosphere of a robot and causes the creation of a certain goal, we then have transformation, not flow, from the infosphere to the goals sphere.

Consequently, an agent architecture is formed from these spheres. The sub-spheres are assumed also to be agents. Figure 6 shows a general view of beliefs, goals, plans, and actions spheres in FM, where the arrows represent flows triggered by outside information.

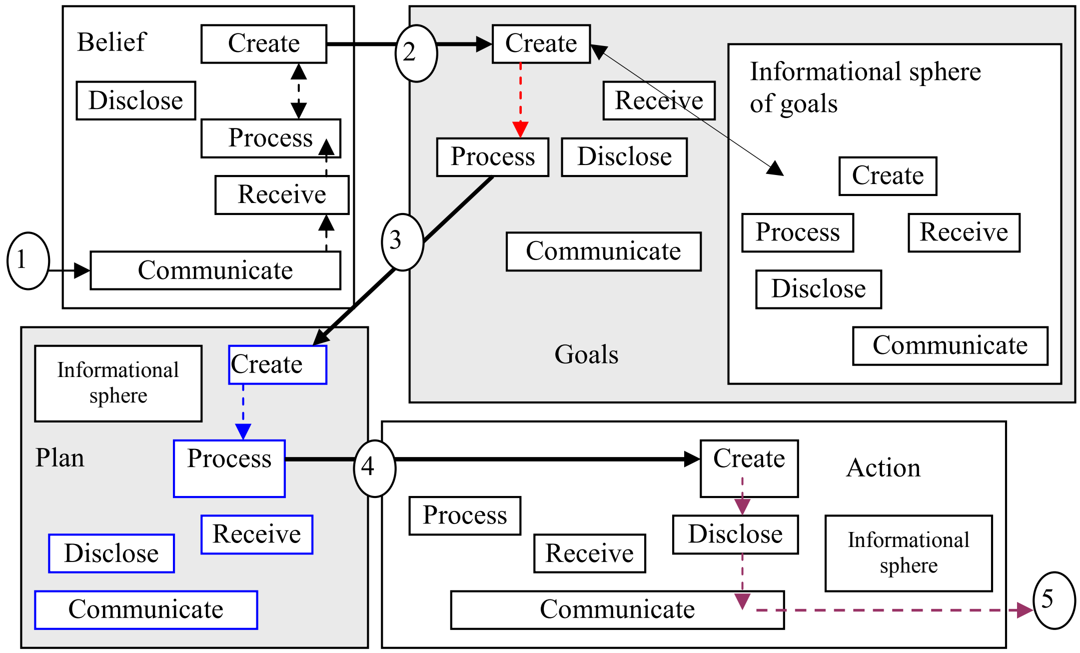

Figure 6. General view of beliefs, goals, plans, and actions spheres in FM, where the arrows represent flows triggered by outside information (point 1).

For simplicity's sake, some internal arrows between stages are not shown. The agent architecture in the figure includes four primary spheres: beliefs, goals, plans, and actions. Beliefs, goals, plans, and actions are considered different species such that the flow does not continue between them. Instead, a flow in one sphere may trigger another flow in another sphere. We call this "transformation" from one type of flow to another.

Consider an outside event in the form of news entering the beliefs sphere (point 1 in the figure). This is an outside piece of information that flows into the beliefs sphere through communication, receiving, and processing stages (dotted lines). It then causes the creation of a new belief. In the figure, the bidirectional arrow between the processing and creation stages indicates several cycles of creation/processing of information/beliefs. The new belief may be processed to create newer belief (multilevel deduction). At the end, the new belief triggers (point 2) the creation of a goal in the goals sphere. Notice that this transformation is denoted by a solid arrow. It does not go between the communication stages, because the information/belief flow is different from the goals flow. Belief cannot flow in the goals sphere, and goals cannot flow in the information sphere.

However, each type of flow can trigger the initiation of a flow in the other sphere. In the figure, we assume that the newly created belief caused the creation of a goal that flows to the processing stage of the goals sphere. Each of the goals, plans, and action spheres has its own information sphere. For example, the information sphere in the goals sphere stores different states, possible goals, etc. This is denoted in the figure by the internal bidirectional arrow between the goals sphere and its informational sphere. It does not go between the communication stages, because the information/belief flow is different from the goals flow. Belief cannot flow in the goals sphere and goals cannot flow in the information sphere. However, each type of flow can trigger the initiation of a flow in the other sphere.

Again, this arrow represents transformation between spheres and not flow of something between them. Upon creation of a goal in the goals sphere, the goal flows to the processing stage, then crosses over (point 3) to the plan sphere in the form of a goal transformed to a plan (point 4). Notice that we are here describing an example of a possible stream of flows. All types of streams are possible, constrained by the five stages' connections in each sphere.

A plan is created in the plan sphere, flows to the processing stage, and is then transformed to the action sphere. There, it triggers an action flow that results in an action in the outside world (point 5).

Figure 7 shows another example where the process is initiated by an outside action.

Suppose that somebody does something that affects the agent (e.g., kicking a robot). The action at point 1 is received in the action sphere and flows to the processing stage of that sphere (dotted arrows). The action sphere may use its informational sphere to decide to activate the belief sphere (later we will add a global agent). This is a transformation from the action sphere to the belief sphere (point 2). In the figure, the action triggers a belief (e.g., under attack) that flows (dotted lines at point 3) to the processing stage, then is transformed (point 4) into the creation of a goal in the goal sphere. This goal flows (point 5) to the processing stage, then triggers (point 6) a plan in the plan sphere. This plan flows to the processing stage (point 7) and then triggers (point 8) an action in the action sphere. The action flow causes a reaction (point 9) for outside action.

5. Scenario

This section describes a version of the so-called "cleaner world" scenario [4]. Its FM description is given in figure 8.

An autonomous cleaning robot has the task of keeping a building clean during the day and guarding it at night. It also monitors its battery and charges it when needed. Therefore, three goals are identified: cleaning, guarding, and monitoring. We declare different spheres, as follows:

- Global agent spheres
- Beliefs agent
- Goals agent with informational sub-agent
- Plan agent with informational sub-agent
- Actions agent with informational sub-agent

Each agent has the five stages schema. Figure 8 shows the design architecture for the cleaning goal. The clock at the top of the figure keeps track of the daytime cleaning period. The start and end of the day are passed to the global agent (point 1A) that processes it and passes the time information to the belief agent.

The belief agent activates (point 2) the cleaning goal at the goal agent. The goal agent activates the plan agent, which in turn triggers the turning on of the action agent.

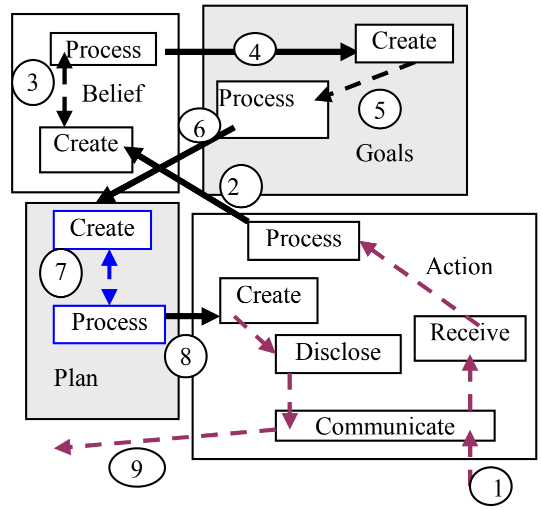

Figure 7. Flows of actions, beliefs, goals, and plans triggered by an outside action at point 1.

For simplicity's sake, the unused stages are omitted. At the action agent sphere, the actual cleaning process is performed with the awareness of the global actions stage. There may be action feedback that is transformed into information that is processed by the informational agent of the actions agent. It is possible that the global agent receives some of this feedback information. The same triggering mechanism occurs at the end of the day to turn off the cleaning goal and turn on the guarding goal.

The black part of the figure represents the global agent that controls other inside agents. In the figure, the global agent interacts with the outside informational environment (to the left) and the outside physical environment (at the bottom); however, these inside agents may communicate with each other without going through the global agent.

The "monitoring goal" is similarly performed by the action agent, as shown in Figure 9. Signals from the battery are received and processed to create necessary actions for charging the battery. The figure shows a sample processing module with a possible pseudo-code. The processing stage of the actions agent has many such modules to perform various action tasks.

6. Plans and goals

The scenario given in the previous section has clarified the nature of actions (e.g., mechanical) that

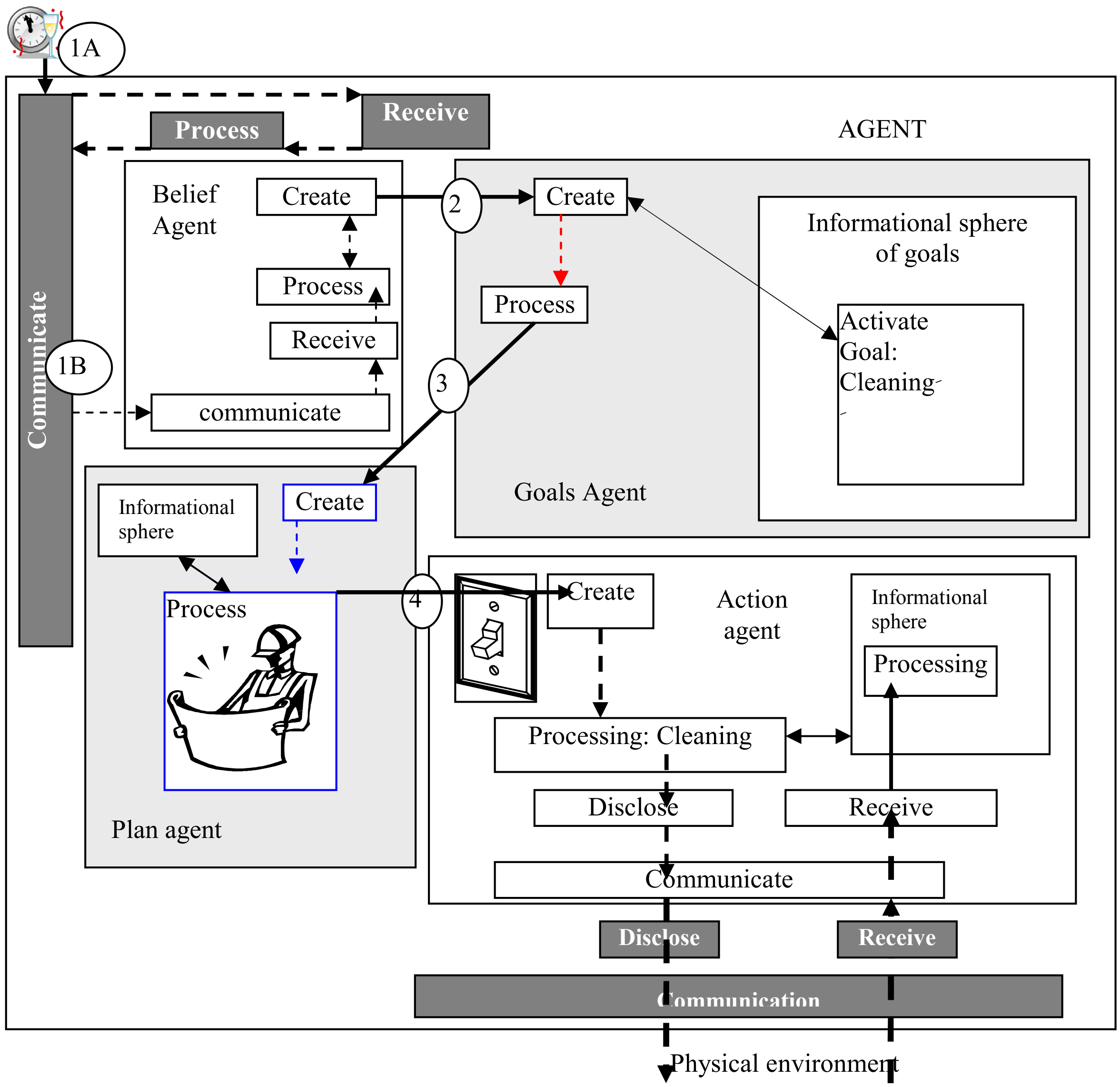

Figure 8. The architecture of the cleaning agent.

are monitored and performed by the agent, such as move and turn off, as modeled in figure 9. The nature of goals and plans in this scheme still need to be clarified.

The modeling of goals is an important aspect in agents design, especially in the requirements analysis phase. In the Belief-Desire-Intention model, desires (goals) are mental states of agents that represent its motivational source of actions. In the literature, goals are usually discussed in terms of types, attributes, and relationships.

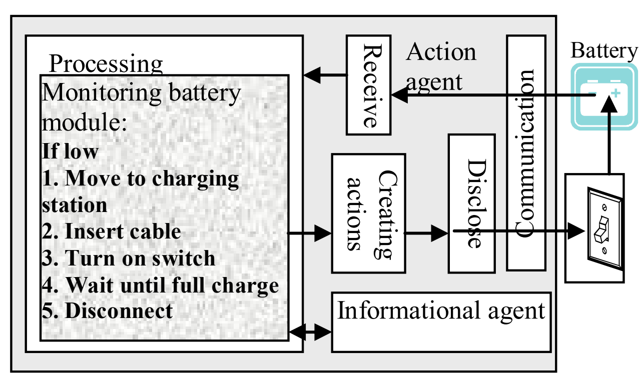

Figure 9. The agent is continuously monitoring.

There are hard goals, soft goals, high-level goals, individual goals, achievement goals, maintenance goals, etc. Goals have name, description, priority, and so forth. Goals may form a hierarchical structure or have an and/or relationship. Active goals trigger the agent to proactively pursue courses of action to achieve the target state. Goal representation in an agent specifies this target state, and the conditions for its creation. Here "creation" means the activation of a goal. In FM, "creation" means new derived goals from previous goals as "information creation" refers to deducing new information from old information (processing-creation stages). In Figure 8, if an obstacle prevents the actions agent from performing a certain action, this is reported to the global agent, who diverts it to the goals agent to identify another goal. The processing of goals is similar to the processing of information. For example, in the Barcelona-New York, example discussed previously, the New Yorker is an agent with the goal of "Checking whether it is raining." This goal includes the sub-goals of "moving to the door," "opening the door," and "moving outside." If the action agent discovers that the door is locked, then the goals agent switches to "finding the key." Such a process is analogous to calculating the truth of a sentence in the information sphere. If it is not possible to find the truth-value of a phrase in the sentence, then it may be possible to replace the phrase with an equivalent phrase.

Goals can be in only one of the five states: created (new), processed, received, disclosed, and communicated. In the human mind, new goals appear as the result of processing in the processing stage, or they may be "generated/created" naturally (e.g., inspiration, revelation). In machines, goals are received goals (e.g., input by designer), or derived from an initial set of received goals, as in deductive logical systems (processing stage).

A goal is also something that can be activated. In Figure 8, the (stored) goal of Cleaning is activated by an information signal originating from the clock at daybreak. It is also deactivated by a similar signal at the end of the day. Goals are also transformed into plans. In Figure 8, guarding is activated by a signal from the informational agent at night. Activating Guarding (of the factory) is transformed into a plan. That is, Guarding triggers initiation of its plan in the plan sphere. In the next section, we illustrate a plan in the context of information security.

7. Information security plan

Interestingly, the flow model can be used to draw a security plan that involves positioning agents at critical points. Suppose that the asset to be guarded is an information system. The system can be visualized in terms of a flow model. It is an information sphere with its own five-stages scheme. An extreme target state (e.g., nonworking hours) can be modeled as having no data being imported to or exported from the system. No data moves (or flows) among the different operational stages during this time. Within each stage, no data is received, processed, created, disclosed, or communicated.

The mission of the guarding agents is to monitor different stages in order to "check" different points in the system. The checks denote the positions of agents in the information flow.

Of course, in actual information systems, policies and procedures may be different at different points. However, regardless of the nature of policies and procedures at each point, their positions are critical. The FM in this case is used to establish a framework for identifying security-significant points where security agents check policies and procedures applied at these points. This approach aims to identify the information stream as the principal focus of security efforts, and it is based on recognizing points of transformation in the flow of information as the hubs around which a security organization is built. It closely emulates people's mental models of protection, through looking for stream of flow (e.g., people, goods) to position guards.

Security agents cover all relevant points of possible breaches of security. Leakage of information can be spotted at the creating, storage, usage, processing, disclosing, and communicating stages and sub-stages. Costs of leakage are different at these points. Security measures can be applied across all stages and to all levels: technological, policy, and human. Security requirements and policy validations can be imposed at the entry to and exit from each stage; thus, it is possible to trace the information flow across stages to identify the location of the security problem. Examples of security measures used by different security agents are the following:

- System authorizations (e.g., wrong destination) as the receiving party of data, recovering from transmission failures
- Anti-virus, hacking watch, firewalls, recovery, monitoring
- Access control
- Prevention of accidental loss of data, updating and maintenance, encryption, backups

8. Conclusion

The aim of this paper is to raise interest in such an approach to agent-oriented applications. The model also needs to be described in some definitional language such as XML or RDF. Its application for agents requires experimental case studies.

As a future work, the flow model can be applied to creation of a security plan that involves positioning of agents at critical points. Security agents cover all relevant points of possible breaches of security. Leakage of information can be spotted at the creating, storage, usage, processing, disclosing, and communicating stages and sub-stages. Costs of leakage are different at these points. Security measures can be applied across all stages and applied to all levels: technological, policy, and human.

9. References

[1] S. Al-Fedaghi, “Modeling Communication: One More Piece Falling Into Place”, The 26th ACM International Conference on Design of Communication (SIGDOC 2008), Lisboa, Portugal on Sept. 22-24, 2008.

[2] S. Al-Fedaghi, “Informational Human-Machine Interaction”, The 2008 IEEE International Conference on Systems, Man, and Cybernetics (SMC 2008), Singapore, 12 - 15 October 2008.

[3] Bratman, M., *Intention, Plans, and Practical Reason,* Harvard University Press. Cambridge, MA, USA, 1987.

[4] Braubach, L., A. Pokahr, D. Moldt, and W. Lamersdorf, “Goal Representation for BDI Agent Systems”, in R.H. Bordini (ed.), PROMAS 2004, LNAI 3346, 2005, pp. 44–65. http://vsis-www.informatik.uni-hamburg.de/getDoc.php/publications/208/goalsemantics_postproc_springer.pdf

[5] Devlin, K., *Logic and Information*, Cambridge University Press, New York, 1991.

[6] FIPA Architecture Board, “Foundation for Intelligent Physical Agents.” http://www.fipa.org/specs/fipa00037/SC00037J.html

[7] Chang-Hyun, J., and J. M. Einhorn, “A BDI Agent-Based Software Process”, *Journal of Object Technology* 4, no. 9, November-December 2005, pp. 101-121. http://www.jot.fm/issues/issue_2005_11/article3

[8] Rao, A., and M. Georgeff, “BDI Agents: From Theory to Practice”, in V. Lesser (ed.), *Proceedings of the First International Conference on Multi-Agent Systems* (ICMAS'95), pp. 312-319. The MIT Press, Cambridge, MA, USA, 1995.