

IEICE **TRANSACTIONS**

on Information and Systems

VOL.E83-D
NO.4
APRIL 2000



A PUBLICATION OF THE INFORMATION AND SYSTEMS SOCIETY

The Institute of Electronics, Information and Communication Engineers

Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3chome, Minato-ku, TOKYO, 105 JAPAN

Formalization of Organizational Intelligence for Multiagent System Design

Behrouz Homayoun FAR[†], *Member*, Hassan HAJJI[†], Shadan SANIEPOUR[†], Sidi O. SOUEINA[†], and Mahmoud M. ELKHOULY[†], *Nonmembers*

SUMMARY Although there are many projects focusing on multiagent systems, there are only a few focusing on systematic design of large scale multiagent system. In this paper we formalize the knowledge representation and sharing of agents, using *symbol structures*, define agencies as *organizations* (i.e., a coalition of agents), propose a formalism to represent *organizational Intelligence*, devise a basic configuration for *generalized agents* (AG), and use them in a large scale multiagent system design. Private knowledge of an AG agent is represented by a *symbol structure* (SS) and AG agents can share their knowledge using combination, specialization and generalization methods that operate on the SS. Opposite to the other works, organizational knowledge, is defined as a property of at least a pair of AG agents. **key words:** *agent, knowledge sharing, organization, symbol structure*

1. Introduction

There are already many projects focusing on agent-based solutions for various scientific and business domains [1], [14], [16], [17]. However, at this moment, there are certain limitations to such solutions, such as, they may not be appropriate for systems and domains with global constraints, domains that rely heavily on knowledge sharing while decisions are made on a local basis, domains with decentralized control and finally domain that require achieving a globally optimal performance.

Building agent-based solutions requires research on at least three topics. First, working on agent theory, the scope and limitations of agent-based solutions. Second, making agent frameworks and infrastructure powerful, interoperable, and secure enough to support large-scale coordinated problem-solving activity. Third, developing new tools to help building agents [2].

In this paper we focus on the second topic. We formalize the knowledge representation and sharing of agents, using *symbol structures*, define agencies as *organizations* (i.e., a purposeful coalition of agents), propose a formalism to represent *organizational Intelligence*, devise a basic configuration for *generalized agents* (AG), and use them in a large scale multiagent system design.

The private knowledge of an AG agent is represented by *symbol structure* (SS) and AG agents can share their knowledge using combination, specialization and generalization methods that operate on the SS (see Sect. 5.2). Opposite to the other works, organizational knowledge, is defined as a property of at least a pair of AG agents.

The structure of this paper is as follows. In Sect. 2 basic issues of multiagent system design are presented and the ways that our project deviates from the main trend of research in this field is clarified. AG agents are introduced in Sect. 3. The organizational intelligence and the way of representing it is introduced in Sect. 4 and the knowledge structure of AG agents is formally defined in Sect. 5. Finally, a conclusion is derived in Sect. 6.

2. Multiagent System Issues

2.1 Knowledge Sharing Issues

The infamous knowledge sharing problem arises from the fact that different systems use different concepts and terms for describing domains. This makes it difficult to take knowledge out of one system and use it in another. Interest in ontologies has grown due to interests in knowledge sharing, data integration, knowledge inter-operation, and reuse [9]. In multiagent system design, ontologies that can encompass agent's internal knowledge as well as the system's organizational knowledge should be developed. Ontology research yet has to produce proposals for how to construct and maintain such ontologies [6].

Knowledge sharing requires knowledge level communication. Various kinds of Agent Communication Language (ACL) have been proposed. ACL is a language with precisely defined syntax and semantics serving as the basis of communication between independently designed and developed software agents. Among ACL languages, Knowledge Query and Manipulation language (KQML), a protocol for exchanging information and knowledge [18] and Knowledge Interchange Format (KIF), a formal syntax for representing knowledge, basically based on first-order predicate logic [10], are the most famous ones.

In our project, we suggest a way of systematically

Manuscript received August 16, 1999.

Manuscript revised November 5, 1999.

[†]The authors are with the Department of Information and Computer Sciences, Saitama University, Urawa-shi, 338-8570 Japan.

building the knowledge base of individual agents and blending it with the ontology of the domain, in a seamless way, using *symbol structures*. Knowledge sharing is treated as revealing the internal symbol structure to the other agents (see Sect. 5.4). In other words, two communicating agents, first share their internal symbol structure and then messages are interpreted. Interpretation is ascribing the same meaning to the constants used in the message. In this way, mutual understanding of the domain constants before further message passing is guaranteed.

2.2 Problem Solving Issues

There are already a number of approaches to distributed problem solving, using agent technology, such as blackboard systems, broadcast methods, delegation, contract networks, multiagent truth maintenance system (TMS) [13], [24], etc. A main goal in multiagent problem solving is to maintain the coherent performance of the agents while maintaining their autonomy. *Cooperation*, *coordination* and *competition* are three major problems in multiagent problem solving. A number of protocols for cooperation, coordination and competition have been suggested (for a survey see [15]). However, there is no single formalism that can cover them all. The *symbol structure*, proposed in this paper gives a coherent view of cooperation, coordination and competition in a distributed problem solving environment (see Sect. 5.4).

2.3 Agent Design Issues

Multiagent system design is a main theme of this paper. There are a number of approaches to agent design, such as: treating agents as objects, agents as expert systems, reactive agents, agents with memory and state [3], etc. There is already an initiative to standardize the agent technology by *Federation of Intelligent Physical Agents*, FIPA (<http://www.fipa.org>).

Scalability is a central issue in multiagent systems design. Although there are many projects focusing on multiagent systems and groupware, it is quite hard to scale them up to include a large number of actors (i.e., agents and/or human counterparts). *Reuse* is another fundamental issue. The once designed agent configurations are to be used in another configurations constantly and repetitively. Existing research projects do not specify how their proposed agent technology can be used for systematic design of large scale multiagent systems. In other words, there is no standard technique for multiagent system design and development, including techniques for requirement analysis, design and implementation.

There are already some agent developing tools available, mostly implemented in Java programming language, such as ABE, JAT and OAA [8]. Such tools

and environments can facilitate the communication (for example, using KQML language in JAT as the knowledge level communication) and message passing (for example, using HTTP and IIOP protocols in ABE and OAA, respectively), but they fail to provide the appropriate aids to build knowledge bases needed to implement symbolic intelligence. The *Generalized Agent* (Sect. 3) proposed in this paper, will help perform such tasks.

3. Generalized Agent (AG)

In this section we introduce *Generalized Agent (AG)* framework to represent and model individual agents. The framework supports human-life epistemology, that is, trying to follow the knowledge representation and problem solving skills of humans. People process and manipulate information in a different way than solution to a mathematical problem or formal proof structure in standard logic. Humans appear to get as input the sensory icons coming from the outside world and developing a coherent model of the situation. Then process rules are used to reason, interpret and explain the situation and derive possible actions. AG framework supports this viewpoint.

3.1 Definitions

Definition 3.1 (Generalized Agent (AG)): Generalized Agent (AG) is an information processing object, acting on the basis of representation, using as input information the sensory icons coming from the external environment, *perceiving*, *conceptualizing*, *interpreting*, and *performing actions* to direct its behavior towards a desired goal.

A basic assumption is that performing these activities is analogous to computation [19], that is, AG acts on the basis of representations and initiates such representations physically as cognitive codes and the actions are carried out as a consequence of operations carried out on those codes.

Among the 4 main tasks mentioned above, *conceptualization* and *interpretation* are studied herewith[†]. Conceptualization is realized by the *symbol structure* (see Sect. 5). Interpretation encounters both interpreting properties of the environment by means of causal laws, qualitative or quantitative model of domain objects, as well as other kinds of *organizational* knowledge, including knowledge due to dynamic coalition (see Sect. 4).

AG agents have the following common characteristics:

Artificiality: AG agents do not exist naturally or biologically in the outside world and they are delib-

[†] Perception and performing action issues are related to sensor organs and motor functions and not studied here.

erately designed by the humans to fit in to their environments.

Adaptability: AG agents have dynamically changing conceptual picture of the situation, and goals, as well as selectivity among a number of mechanisms to cope with the changing environments.

Rationality: AG agents are rational. Given a set of actions, the AG selects an action which is feasible and optimal.

Individual Intelligence: AG agents have efficient problem solving and decision making skills on a personal basis, such as segmentation of the search space and jumping to conclusion.

Communication and Knowledge Sharing: AG agents have communication and knowledge sharing skills that help their individual problem solving and decision making skills be utilized efficiently.

Memory and State: AG agents have memory and state. Their profile of actions and states over time is called *history*.

Definition 3.2 (Organization): An organization is a goal directed coalition of agents in which the agents are engaged in one or more tasks and knowledge and capabilities are distributed among the agents.

Participating agents in an organization are described by either of the followings.

1. *Class*: Specifying inner environment.
2. *Interface*: Specifying the input/output and/or functions.

AG agents are assumed to be goal seeking open loop artifacts interacting with the environment. Fulfillment of purpose or adaptation to a goal involves a relation among three terms:

- The purpose or goal;
- The character of the artifact (inner environment);
- The environment in which the artifact performs (outer environment);

The inner environment is an organization of parts (see Fig. 1) capable of attaining the goals in some range of outer environments. Describing AG agents by their *class* means specifying their *internal implementation*.

On the other hand, there may be a number of functionally equivalent agents capable of attaining the goals in the same outer environment. The resemblance of the behavior without identity the inner environment is particularly feasible if the aspects in which we are interested arise out of the organization of the parts, independently of all but a few properties of the individual components. [23]. Describing AG agents by their *interface* means encapsulating (i.e., black boxing) the actual

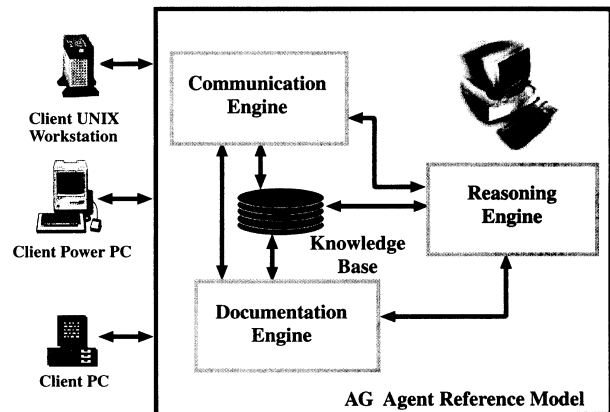


Fig. 1 Internal structure of a generalized agent.

internal implementation.

Another way to make performance level of AG agents get close to that of human being is providing them with *insights*.

Roughly speaking, insight is defined as the ability of deriving information from a set of data through methods other than monotonic inference based on causal assumption. In expert systems, insights are defined in terms of shallow rules that ease obtaining a solution and deep rules that allow more precise reasoning. Yet the relation between the deep knowledge on underlying mechanisms and heuristic rules is always vague [26]. We define insights in terms of inferential activities enabling to perform *generalization* and *specialization* from a given set of data (see Sect. 5.2), and devise rules for generalization and specialization within the symbol structure (SS) framework.

3.2 Reference Model

A number of reference models for building (intelligent) agents have already been proposed. For instance, the Agent Reference Model, specified by *Federation of Intelligent Physical Agents*, FIPA (<http://www.fipa.org>) defines a framework within which agents exist and operate. FIPA's reference model includes: *Agent Communication Channel (ACC)* to route messages between agents within the platform and to agents resident on other platforms; *Directory Facilitator (DF)* to provide directory service for the agents by storing descriptions of the agents and the services they offer; *Agent Management System (AMS)* to manage the creation, deletion, suspension, resumption, authentication and migration of agents on the agent platform; *Agent Platform Security Manager (APSM)* to maintain the intra- and inter-domain security policy; and *Internal Platform Message Transport (IPMT)* to exchange messages within the platform. The Agent Reference Model does not specify the internal structure of the agents.

Figure 1 shows the reference model for GA agents

in our project. Similar to conventional expert systems, each agent has a *local knowledge-base* and a *reasoning engine*. Compared to the conventional expert systems, a main difference is that all agents have an additional *communication engine* and a *documentation engine*.

AG's reasoning engine includes,

- Patterns for actions that AG agent actually performs.
- Rules of performance (i.e., maps) for dynamically generating new patterns.

Patterns are triggered upon receiving a message either from the communication and documentation engines or another agent. Each pattern has a name, a number of input parameters and output parameters. These together are called a *signature* and the set of all signatures of an AG is its *interface*.

The communication and documentation engines facilitate handling messages and run patterns. The communication engine is mainly responsible for maintaining connection to the network, communicating with other agents and managing messages. A three step protocol for managing messages, agent identification, query processing and payment processing, is devised.

Documentation engine performs tasks, such as, acquiring data from the other agents, as requested by the reasoning and communication engines, and preparing and reformatting data items to be appropriate for transferring over the network.

Documentation and communication engines, facilitate agents' local knowledge being shared with the other agents.

4. Organizational Intelligence (OI)

Organizations, of various forms, physical, cognitive, temporal and institutional have been studied in operation research, management and computer sciences. The game theoretic approach to study organization focuses on modeling and suggesting computational algorithms for certain aspects of the coalitions, such as social welfare [21], individual rationality, voting consensus, etc. The computational approach focuses on identifying general principles of organization and their exceptions. The proposed theories extend the information processing capabilities of individual agents to an organization level, through defining concepts such as, *bounded rationality* [22]. There is another approach focusing on environment centered analysis and design of coordination mechanisms [7][†].

The already proposed multiagent systems, as a purposeful organization of agents, do not offer global optimization for the knowledge resources and skills of the participant agents, i.e., agent organization's capabilities in almost all of the cases is less than the sum of capabilities of its participant agents. However in many

natural coalitions, such as bees and ants societies, the organization has the merit of overcoming shortcomings of participant agents.

We believe that a main reason is the improper interpretation and implementation of the *Organizational Intelligence* (OI). Here we briefly discuss the OI features, its underlying assumptions and the model will be introduced in the next section.

4.1 Intelligence of Pair (IoP)

There is a fundamental question whether OI resides in an agent itself, external to the agent, or an outcome of the interaction among agents.

At the first glance, it seems that OI includes knowledge distribution (who knows what?) and sharing (how it can be utilized?) as mentioned in [5]. All of the proposed theories and formalisms have implicitly assumed that OI exists and implemented either inside a single agent or external to it using a meta-agent (e.g., directory and ontology service agents). However there are certain difficulties in both logical formulation and actual implementation of such theories. This is mainly due to ignoring the dynamic interactions among the agents when devising the components of OI.

A novel point proposed in this paper is that in a purposeful (i.e., not random) organization, OI is a property of interaction among agents and can only be ascribed to at least a pair of agents. We call this '*Intelligence of Pair (IoP)*' assumption.

4.2 History of Patterns (HoP)

In biological coalitions, participants may have a kind of *role* or *function* (during interaction with the other participants), if they show some persistence in their profile of actions over time. The same could be devised for artificial coalitions. As a matter of fact, it is not difficult to find organizations that display non-random persistent and repeated patterns of actions [5].

Agents act and perform in a physical world. Their past experiences can be recorded and explained in terms of their *histories*, that is, their *profile* of actions and *states* that they go through.

Intuitively, histories can display certain patterns. A basic feature of state representation is that it assigns a certain characteristic to its reference agent. Therefore it is possible to define OI patterns with reference to agents' history.

Another novel point proposed in this paper is that OI patterns emerge from discovering a persisted state or an ordered pattern in the agent's profile of actions. We call this '*History of Patterns (HoP)*' assumption.

In biological coalitions, persistence is considered to

[†]For a survey on computational organization theories see Ref. [4].

be the most interesting characteristic and is believed to be governed by natural selection law. In artificial coalitions, besides persistence, other kinds of ordered patterns, such as repetition cycles, may also be considered as important features of the coalition.

IoP and HoP assumptions account for dynamic interactions and a computation method based on this assumption is proposed in Sect. 5.5.

5. Symbol Structure (SS)

In this section we introduce *Symbol Structures* (SS) which is used to represent private knowledge of an individual agent and help derive organizational knowledge of the agency. The SS model is multi-level graph. Formal rules of inter-layer transition are introduced. Flexibility, extendibility and interoperability are three main advantages of knowledge representation and reasoning with SS.

Species of information processors, including AG agents, are members of the family of symbol systems. A symbol system is a machine that, as it moves through time, produces an evolving collection of symbols. One of the capabilities that all kinds of symbol systems have in common is that they act on the basis of symbolic representations [23].

SS is the internal representations of the knowledge of the agent about the outer environment to which the agent is seeking to adapt. It is composed of *concepts*, and *relations* among concepts (See Fig. 2). SS is a way to assimilate separate facts into a coherent image that with the aid of some logical inference rules, can be used to detect contradictions and draw decisions.

They possess a common symbol set and a number of *actions* that operate upon the SS. Knowing the common symbol set and actions, together with the assumption that the problem solving behavior is concerned with the representations by certain general principles, makes it possible to explain an important segment of the regularities in problem solving behavior of those agents.

5.1 Definitions

We assume that the perception mechanism of the agent gets as its input the sensory icons coming from the external world and produces at its output a symbol structure[†]. We formalize herewith the information processing technique by means of symbol structures.

Definition 5.1 (Symbol Structure): A symbol structure (SS) is a finite connected multi-layer bipartite graph. There are two kinds of nodes in each layer of SS: *concepts* (c) and *relations* (r). The concepts belong to concept set $C = \{c_1, \dots, c_n\}$ and are linked by relations belonging to relation set $R = \{r_1, \dots, r_m\}$ to form a model that approximates an entity or scene e , which is represented by the concepts from the set C

and relations from the set R .

One source of difficulty when processing concepts, is distinguishing a concept at various levels of abstraction, as well as differentiating between generic concepts and their instances. Function *type* is defined to ease such differentiation. Henceforth, we basically work with the types of concepts rather than individual concepts. We will further define a type hierarchy which is a higher order relation between various levels of abstraction. Relations, are also classified by types.

Definition 5.2 (Type): The function *type* maps concepts (C) and relations (R) onto a set T . The elements of T are called type labels.

$$type : C \cup R \rightarrow T \quad \forall x \in C \cup R \mid type(x) \in T \quad (1)$$

Concepts c_1 and c_2 are of the same type if, $type(c_1) = type(c_2)$. Relations r_1 and r_2 are said to be of the same type if, $type(r_1) = type(r_2)$ and r_1 and r_2 have the same number of arcs.

Definition 5.3 (Instance): Instances of a concept c , shown by $I(c)$, are every occasions that c comes to existence. It is shown by:

$$I(c) : \{c^1, c^2, \dots, c^j, \dots\} = (\forall j, c^j) \quad (2)$$

Similarly, instances of a type t , shown by $I(t)$, are every occasions that every concept c of type t comes to existence.

Definition 5.4 (Subtype): The subtype t_s of type t is the set of all instances of type t_s which also are instances of type t .

$$\forall i, j \quad (c_i^j \in I(t_s)) \wedge (c_i^j \in I(t)) \rightarrow t_s \prec t \quad (3)$$

(\prec) is a two point symbol denoting subtype relation;

Type hierarchy provides a means of evaluating a concept at various levels.

Definition 5.5 (Type Hierarchy): The type hierarchy is a partial ordering defined over the set of type labels, T . The symbols (\prec) and (\succ) designate ordering. For the type labels, s , t and u , if $s \prec t$, then s is called a subtype of t ; or t is a supertype of s , shown by, $t \succ s$.

An example is shown in Fig. 2. Rectangular nodes, such as *Bank*, *Account*, etc., are concept types and circular nodes, such as *Has*, *Holds*, etc., are relation types. The *Account* has *Credit_Account* and *Cash_Account* as its subtypes. Any instance of *Account*, such as an account belonging to an individual x denoted by *Account:-x* is an *instance*.

5.2 Reasoning

Until now we have developed the basic syntax of SS.

[†]In this paper we do not encounter the way of building symbol structures. We assume that the knowledge of the AG is already available in the SS form.

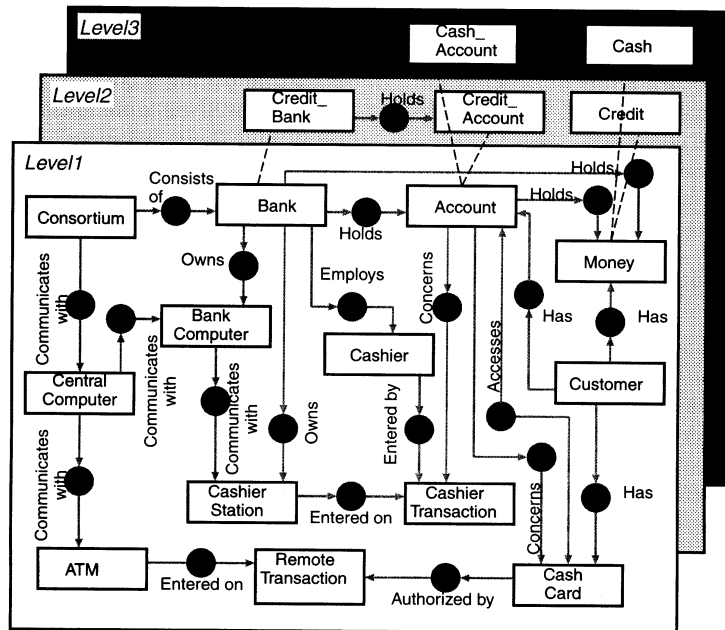


Fig. 2 Example of a triple-layer SS.

This is powerful enough to represent knowledge at various levels of abstraction. In this section we show how to manipulate the knowledge and reason with symbol structures.

Basically, there are only two kinds of reasoning rules, *combination* and *insight* rules. Combination rules are the followings:

Join rule: Join rule merges identical concepts. If a concept c in u is identical to a concept d in v , then let w be the symbol structure obtained by deleting d and linking to c all arcs of relations that had been linked to d .

Simplification rule: Redundant relations of the same type linked to the same concept in the same order can be reduced by deletion all but one. If the relations r and s in the symbol structure u are duplicates, then one of them may be deleted from u together with all its arcs.

An *insight* is defined as the ability to *generalize* and/or *specialize* information from a symbol structure. Specialization is equivalent to increasing the precision in modeling. The outcome of applying specialization rules on a SS is another SS which is more restricted than the original one. Conversely, generalization proceeds in the reverse order of specialization.

Generalization/Specialization rule: For two arbitrary levels u and v of any symbol structure, if u is identical to v except that some type labels of the nodes of v are restricted to subtypes of the same nodes in u , then u is called a specialization of v , written $u < v$, and v is called a generalization of

u .

Generalization defines a partial ordering among the levels of any symbol structure, called the generalization hierarchy. Specialization is equivalent to replacing the type label of a concept with the label of a subtype. For three arbitrary levels of any symbol structure u , v and w , the following properties hold.

- Transitive:

$$(u < v) \wedge (v < w) \rightarrow (u < w) \quad (4)$$

- Antisymmetric:

$$(u < v) \wedge (v < u) \rightarrow (u = v) \quad (5)$$

Note that specialization rule is not a rule of causal inference. It only enforces selectional constraints by preventing certain combinations from being derived. Obviously, opposite to specialization, generalization preserves truth but does not necessarily preserve selectional constraints.

An important feature of SS is that, two different symbol structures may have a common generalization (and/or specialization).

5.3 Actions and States

There are finite sets of *actions* and *states* associated with each agent. Given a set of all possible actions, A , an agent's action is a subset $B \subseteq A$.

Both *actions* and *states* take concepts and relations as their attributes. For example, moving money

from one account, say x , to the other, say y , is associated with an action MOVE, where,

$$\text{MOVE}(\text{Money}, \text{Account}:-x, \text{Account}:-y)$$

means that action MOVE works on three concepts, a generic type *Money* and two instances *Account:-x* and *Account:-y*.

Similarly, the *state* of the world after performing this action is

$$S_1(\text{Money}, \text{Account}:-y)$$

5.4 Interaction among Agents

Now we have a framework for representing and reasoning with the knowledge on an individual agent basis. Moving from any level of abstraction to another is supported. Knowledge sharing by moving from one agent to another and on an organizational basis requires defining the basic agent interactions, i.e., *cooperation*, *coordination* and *competition*. For a pair of agents to interact, each should maintain a model of the other agent, as well as a model of future interactions [15].

Cooperation: Cooperation is revealing an AG agent’s *goal* and the knowledge behind it, i.e., its symbol structure to the other party. In cooperation both agents have a common goals.

Coordination: Cooperation is revealing an AG agent’s *goals* and the knowledge behind it, i.e., its symbol structure to the other party. In coordination, agents have separate goals.

Loose Competition: Loose competition is revealing only an AG agent’s *goals* but masking the knowledge behind it to the other party.

Strict Competition: Strict competition is neither revealing an AG agent’s *goals* nor the knowledge behind it to the other party.

Knowledge sharing is equivalent to merging two or more symbol structures using *combination* and *insight* rules. Knowledge sharing is fully utilized in the case of *cooperation* and *coordination*. In case of loose or strict competition the SS model of the other agent should be predicted based on its observable states. This is a topic yet to be studied.

5.5 Computational OI

Here we propose a computation method for generating OI concepts based on the IoP and HoP assumptions (see Sect. 4). Figure 3 depicts the idea. In this method, first, a pair of AG agents are selected and by using reasoning rules (see Sect. 5.2) their pairwise profile is produced (see Sect. 5.4). Then by using a simple pattern detection algorithm, possible repetition and persistence patterns are derived and added to the knowledge base of the organization.

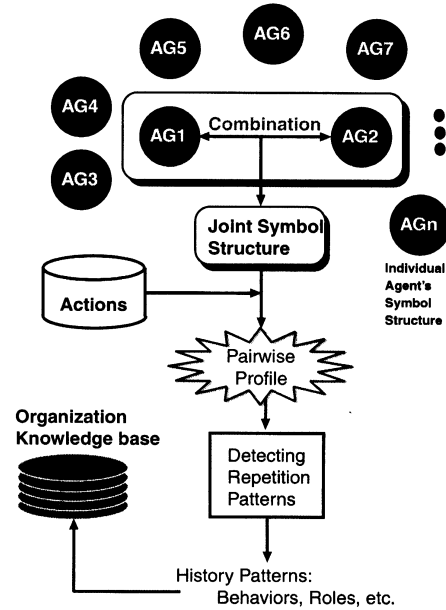


Fig. 3 Organization intelligence concept.

Definition 5.6 (Agent’s Profile): An agent’s profile, \mathcal{P} is a finite sequence of (s, a) , recorded during reasoning on the symbol structure. where, s is problem solving state; a is action;

$$a \in B; \quad B \subseteq A;$$

B is the action set available to agent and A is the set of all possible actions.

Definition 5.7 (Pairwise Profile): A pairwise (i.e., joint) profile of two agents, i and j , \mathcal{P}_{ij} is a finite sequence of $([s_i, s_j], [a_i, a_j])$, during reasoning on the symbol structure, after merging their symbol structures. Each action is a pair of joint actions of the two agents.

$$\forall [a_i, a_j] \quad a_i \in B_i, \quad a_j \in B_j$$

$B_i \subseteq A$ and $B_j \subseteq A$ are the action set available to agent i and j , respectively.

Deriving *organizational knowledge* based on IoP and HoP assumptions has many advantages: first, concepts derived in this way can be explained in terms of organization and its comprising agents without reference to any other intermediary concepts. Second, it provides a framework for comparing and evaluating completely different organizations. Third, the organizational knowledge base is updated dynamically, accounting for different configuration of the participant agents. Finally, it is an appropriate vehicle to explain the need for services of a certain agent in an organization. All of these factors are necessary in organization design.

6. Discussion and Conclusion

In this paper we formalized the knowledge representation and sharing of agents through devising, representing and incorporating organizational knowledge in multiagent system design. A distinguishing point was attributing the organizational knowledge to at least a pair of AG agents. The private knowledge of an agent is represented by *symbol structure* and agents can share their knowledge using combination, specialization and generalization methods. We gave a coherent view of agent interaction, i.e., cooperation, coordination and competition.

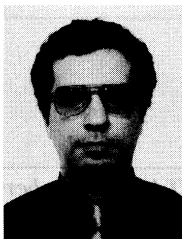
There are certain advantages of using Symbol Structure (SS) to represent knowledge of AG agents. Among them, we can mention that SS is semantically richer than semantic networks and rule-based representations because both the concepts and relations are augmented with *types*. Furthermore, in this formalism, as the actions take concepts and relations as their attributes, flexibility in representing concepts in SS will increase the probability to achieve a goal.

Applications using the framework and techniques described in this paper, such as a multiagent system for electronic commerce [8], a multiagent distributed fault diagnosis system for wide area networks [11], a multiagent intelligent tutoring system (ITS) system [12], a computer aided software engineering (CASE) tool for object oriented software design (OOExpert) [20] are under investigation and development.

Dealing with uncertainty when predicting the SS model of the other agents in case of loose or strict competition and blending SS with the *belief networks* is a future research topic.

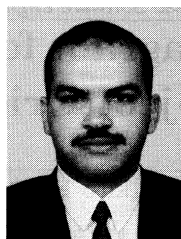
References

- [1] J.M. Bradshaw, ed., *Software Agents*, p.480, MIT Press, 1997.
- [2] J.M. Bradshaw, M. Greaves, H. Holmback, T. Karygiannis, B. Silverman, N. Suri, A. Wong, and W. Jansen, "Agents for the Masses," *IEEE Intelligent Systems & Their Applications*, vol.14, no.2, pp.53-63, 1999.
- [3] W. Brenner, R. Zarnekow, and H. Wittig, *Intelligent Software Agents*, Springer Verlag, Berlin, 1998.
- [4] K.M. Carley and M.J. Prietula, *Computational Organization Theory*, Laurence Erlbaum Associates, Hillsdale, NJ, 1994.
- [5] K.M. Carley and L. Gasser, "Computational organization theory," in *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, ed. G. Weiss, pp.299-330, MIT Press, 1999.
- [6] B. Chandrasekaran, J.R. Josephson, and V.R. Benjamins, "What are ontologies, and why do we need them?" *IEEE Intelligent Systems & Their Applications*, vol.14, no.1, pp.20-26, 1999.
- [7] B. Decker, "TAEMS: A framework for environment centered analysis and design of coordination mechanisms," in *Foundations of Distributed AI*, eds. G.M.P. O'Hare and N.R. Jennings, John Wiley & Sons, 1996.
- [8] B.H. Far, S.O. Soueina, H. Hajji, S. Seniepour, and A.H. Hashimoto, "An integrated reasoning and learning environment for WWW based software agents for electronic commerce," *IEICE Trans. Inf. & Syst.*, vol.E81-D, no.12, pp.1374-1386, Dec. 1998.
- [9] A. Farquhar, R. Fikes, and J. Rice, "The ontolingua server: A tool for collaborative ontology construction," *Knowledge Systems Laboratory, KSL-96-26*, 1996.
- [10] M.R. Genesereth and R.E. Fikes, *Knowledge Interchange Format Version 3.0 Reference Manual*, The DARPA Knowledge Sharing Initiative, 1992.
- [11] H. Hajji, B.H. Far, and Z. Koono, "Intelligent agents for probabilistic diagnosis-restoration planning in WAN," *IEICE Technical Reports*, AI98-56, 1998.
- [12] A.H. Hashimoto, H. Hajji, and B.H. Far, "A probabilistic approach for student modeling task," *13th Annual Conference of Japanese Society for Artificial Intelligence, JSAI'99*, pp.68-69, 1999.
- [13] M.N. Huhns and D.M. Bridgeland, "Multiagent truth maintenance," *IEEE Trans. Syst., Man. & Cybern.*, vol.21, no.6, pp.1437-1445, 1991.
- [14] M.N. Huhns and M.P. Singh, *Readings in Agents*, Morgan Kaufmann Publishers, 1997.
- [15] M.N. Huhns and L.A. Stephens, "Multiagent systems and societies of agents," in *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, ed. G. Weiss, pp.79-120, MIT Press, 1999.
- [16] N.R. Jennings, K. Sycara, and M. Wooldridge, "A roadmap of agent research and development," *Autonomous Agents and Multiagent Systems*, vol.1, pp.7-38, 1998.
- [17] N.R. Jennings and M. Wooldridge, eds., *Agent Technology, Foundations, Applications and Markets*, Springer-Verlag, 1998.
- [18] *DRAFT Specification of the KQML Agent — Communication Language*, The DARPA Knowledge Sharing Initiative, 1997.
- [19] Z.W. Pylyshyn, *Computation and Cognition, Towards a Foundation of Cognitive Science*, MIT Press, 1989.
- [20] R.S. Wahono and B.H. Far, "OOExpert: Distributed expert system for automatic object-oriented software design," *13th Annual Conference of Japanese Society for Artificial Intelligence, JSAI'99*, pp.456-457, 1999.
- [21] T.W. Sandholm, "Distributed rational decision making," in *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, ed. G. Weiss, pp.201-258, MIT Press, 1999.
- [22] H.A. Simon, *Administrative Behavior*, Free Press, New York, 1947.
- [23] H.A. Simon, "Cognitive science: The newest science of the artificial," *Cognitive Science*, vol.4, pp.33-46, 1980.
- [24] R.G. Smith, "The contact net protocol: High level communication and control in a distributed problem solver," *IEEE Trans. Comput.*, vol.C-29, no.12, pp.1104-1113, 1980.
- [25] J. von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*, John Wiley & Sons, 1944.
- [26] K. Yoshida and H. Motoda, "An approach to hierarchical qualitative reasoning: Constructing shallow knowledge from deep knowledge," *Journal of Japanese Society for Artificial Intelligence*, vol.4, no.4, pp.447-455, 1990.



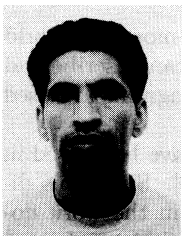
Behrouz Homayoun Far received BSc. and MSc. degrees in Electronic Engineering in 1983 and 1986, respectively, from Tehran University, Iran. He has received his Ph.D. degree from Chiba University - Japan, in 1990. He is currently an Associate Professor at the Department of Information and Computer Sciences, Saitama University - Japan. The research fields of his interest are qualitative reasoning, automatic programming and distributed AI.

Dr. Far was a secretary of SIG-KBSE/IEICE from 1995 to 1997. He is a member of the ACM, IEEE Computer society, Japanese Society for Artificial Intelligence, and Information Processing Society of Japan.



Mahmoud M. Elkhoully received BSc. and MSc. degrees from Helwan University-Egypt in 1987 and 1994, respectively. In 1995 he has received his second MSc. degree from Cairo University in Computer sciences. He has received Ph.D. from the Department of Information and Computer Sciences, Saitama University in March 2000. His research interests are ICAI, Information Retrieval, and Automatic Programming.

Dr. Elkhoully is a member of the IEEE computer society and Japanese Society for Artificial Intelligence.



Hassan Hajji received BSc. in Mathematics in 1995, from Mohamed Premier University - Oujda, Morocco, and MSc. in computer science, from Saitama University, Japan in 1999. He is currently a Ph.D. candidate at the Department of Information and Computer Sciences, Saitama University. His research interests are in distributed network fault and performance management.



Shadan Saniepour received BSc. in Electronics and Telecommunication in 1993, from Iran University of Science and Technology, and MSc. in 1999 from Saitama University, Japan. She is currently a Ph.D. candidate at the Department of Information and Computer Sciences, Saitama University. Her research interests are in Case based Reasoning, Data mining and Distributed AI.



Sidi O. Soueina received BSc. and Msc. in 1993 and 1997 from Almustansiriyah University - Iraq, and Saitama University - Japan, respectively. He has received Ph.D. from the Department of Information and Computer Sciences, Saitama University in March 2000. His research interests are Distributed Artificial Intelligence, Machine Learning and Reasoning Under Uncertainty.