Revised version of the paper published in:
Proceedings of the XI Brazilian Symposium on Artificial Intelligence
SBC, Fortaleza, October 17-20 1994. p.535-546.

On Entering an Open Society

Antônio Carlos da Rocha Costa
Jomi Fred Hübner
Rafael Heitor Bordini
{rocha,jomi,bordini}@inf.ufrgs.br

Instituto de Informática - UFRGS
Caixa Postal 15.604
90.501-970 Porto Alegre, RS. BRASIL
[caracol.inf.ufrgs.br:pub/inf.ufrgs/ia]

Abstract

This paper concerns the problem of agent migration between open societies. In particular, it focuses on the problems of an agent entering an open society. Two functional levels in open societies are recognized, the language and interaction level, and the knowledge and performance level. It is argued that a functional approach to those levels, permitting that social roles and social processes be defined in each of them, can be handy in the uniform treatment of the various problems involved in entering a society. In particular, it is argued that the functional approach allows the reduction of most of such problems to the problems of recognizing, learning and negotiating roles and protocols in social structures, which are problems that can be treated in a general form, from the point of view of that approach. Some mechanisms for roles identification are proposed and discussed.

1. Introduction

This paper deals with some aspects of distributed AI systems [DEM 90]. In particular, it deals with open multi-agent systems, here called open societies. The problem of agent migration between open societies is considered in its most critical aspect, namely, the entrance of an agent in a new society. Of the troubles and benefits brought by such operation to the agent and the society, we focus on the troubles that the agent must solve in order to adequately establish itself in the functional structure of the society.

The paper consists of two main parts. The first part has a general character. It comprises sections 2 to 5. The second part, sections 6 and 7, deals with entrance problems in a particular sample society. In section 2 we present the concept of open society, and we discuss general aspects of agent migrations between open societies. In sections 3 and 4 we consider problems encountered by agents and societies when agents enter or leave societies. In section 5 we suggest that most of those problems can be seen as belonging to two types: problems of language and interaction, and problems of knowledge and performance. We suggest that the two kinds of problems be seen as refering to two functional levels in open societies, so that they can be put under the general functional framework studied elsewhere

[COS 93a] [COS 93b] [COS 94] and described briefly here. In particular, the problem of role recognition can be seen as common to both functional levels.

A sample open society, the Producer-Intermediary-Consumer society, already considered in a simplified version in [COS 93b], is examined from the point of view of the entrance of new agents into it. We consider the problem role recognition in such sample society, and we summarize work going on in [HÜB 94] dealing with this problem in that sample society. In the Conclusion we comment another work also going on concerning agent migration between open societies [BOR 94].

2. Open societies and agent migration

Open societies are societies that agents can enter or leave freely. Also, they are societies where agents can move freely from one site to another. We call such movements **agent migrations**. A migration is an **external** migration if the agent moves between two societies. It is an **internal** migration if the agent moves within its society, between two sites.

External and internal openness are two independent properties of a society, which may exist in any combination in a society. A fully open society is a society which is both internally and externally open. If an agent moves from a society **s** to a society **s'** (in an external migration), or from site **s** to site **s'** in its current society, then we say that **s** is the **leaving point** and **s'** is the **entering point** of the agent's migration.

External migration may occur for two reasons. Firstly, the agent may need some social functions that is only available in another society, different from that in which it currently is, so that it has to migrate to be able to benefit from that functions. Secondly, a society may need agent capabilities that can only be performed by an agent who resides in another society, so that the agent has to migrate, for the society to be able to benefit from the agent's capabilities. Internal migration occurs for the same reasons as external migration, only that the needs do not refer to societies, but to sites in a given society.

Internal and external migrations can be formally treated as processes of the same kind, since the types of procedures involved are the same. The differences between them concern the concrete types of functions and agent capabilities involved, since internal migrations may be performed by specialized, non-autonomous agents, while external migrations may demand full autonomy from the migrating agent. In the sequel, we focus on migrations that can be characterized as external migrations, involving fully autonomous agents.

3. Entering and leaving a society

When an agent enters a society, a few problems are posed for both the agent and the society, which must be solved before the agent and the society can benefit from each other. In [BER 92] some of those problems were analysed, and solutions were proposed. That paper concerned mainly problems of knowing each other capabilities, that is, the agent and the society (i.e., the other agents, that were already in the society before the migration) must know each other's capabilities before they can start to cooperate effectively.

We think, however, that such kind of problem is one of a set of problems that need some further clarification. We identify two main types of problems that agents and societies must solve, in order to carry out successful migration procedures. We also identify two main types of solutions for each of such problem types. We think that migration problems should be classified as either **problems of the societies**, or as **problems of the migrating agent**.

The main problem that a society must solve when it is entered or left by an agent it that of maintaining its **functional integrity**. That is, the society must assure that after the agent migration it continues functioning as well as (or better than) it was functioning before the migration. The society that is left by the agent must be able to find among the remaining agents some that can become responsible for the functions that the leaving agent was responsible for, before it left.

The society that is entered by the agent must be able to find ways for adequately placing the agent within the existing social structure, and must also be able to modify such social structure, so that the agent can perform the functions that are being required from it. Specialized means should be available to open societies, to enable them to solve the problem of keeping their functional integrity in view of agent migrations. This is partially the subject of on going work [COS 94] [HÜB 94].

The main problems of a migrating agent are problems of helping the societies it is leaving or entering, to keep their functional integrity. There are problems of helping the society it leaves to transfer to the reamining agents the functions which it was responsible for, before the migration, and problems of helping the society it is entering to modify its structure, so that the agent can become responsible for functions in that society.

4. Problems in entering societies

We consider, in this section, problems that an agent must face when it enters a new society. We assume that only open societies can be entered or left by migrating agents, so that if an agent enters or leaves a society, the society is prepared to solve the problems produced by that entrance or departure. In the same way, we assume that an agent enters or leaves a society only after being sure that it is capable of dealing with the types of problems that it will face when doing that, so that the agent is prepared to solve the problems produced by that entrance or departure.

This means that we consider only migrations that have been negotiated between the migrating agent and the two involved societies, and that require the development of no new mechanism in either the agent or the societies. In this situation, we can consider that the problems that the agent must be involved in, when entering a society, can be classified as either problems of **language and interaction** or problems of **knowledge and performance**. We see that the possible solutions to such problems can be classified as either statically **pre-programmed solutions**, or as solutions that are dynamically **constructed solutions**, but obtained through pre-defined mechanisms. This will be detailed below.

Problems in the level of language and interaction concern the ways the agent will behave in order to properly place itself within the social structure of the society it has just entered. They are problems of how to use language expressions, and of how to organize interactions, in the new society. Problems of knowledge and performance concern the ways the agent will behave in order to properly perform the functions that the society wants it to perform, and also to properly benefit from the functions that are performed in the society by the other agents.

Problems of language and interaction, and problems of knowledge and function are not unrelated. On the contrary, their solutions should probably be carried on simultaneously, for each can help in obtaining the other. However, they concern different features of an agent, features that can be even architecturally distinguished by being implemented by different agent modules, so the division is conceptually interesting.

5. The functional approach to the entrance problem

The problems that agents must face when entering a society - problems of language and interaction, and problems of knowledge and performance - show a common structure when seen from a functional point of view where agents and their environments are organized in terms of social roles and social interactions [COS 93a], [COS 93b].

From such point of view, concerns of both language and interaction, and knowledge and performance of agents, can be seen as sets of social processes and structures, where agents perform some roles, as will be explained below.

In [POP 93], agent roles were identified to agent behaviors in problem solving activities, and social processes were seen as communication protocols that agents were expected to follow when they were interacting. This approach puts together a feature that concerns the language and interaction type of problem with a feature that concerns the knowledge and performance type of problem. We propose that both types of problems should be treated in terms of social processes and social roles. That is, both in communication and interaction, and in knowledge and performance, processes involving several agents are present, and in each such process agents may perform different roles. For instance, both in communication and interaction, and in knowledge and performance, different agents are responsible for different roles, as initiators or terminators of processes, as sources or sinks of information, as target or sources of actions, as coordinating or coordinated agents, etc.

A general approach to functionality, as proposed in [COS 93a], [COS 93b] allows one to see those types of problems as concerning two different functional levels in a society, the level of language and interaction, and the level of knowledge and performance. From such point of view, both levels can be organized under the same principles - social processes and social roles - so that solutions that apply to one of them will possibly apply to the other.

In particular, in [POP 93], the problem of knowing each other better was analysed, and some solutions for it were proposed. This problem can be equally identified in both functional levels: agents must know each other better not only concerning their respective knowledge and functional capabilities, but also with respect to their language and interaction capabilities. In this respect, it is remarkable how usual it is to assume, in distributed AI, that not all agents know everything or that not all agents are capable of performing the same way in a society, concerning the problems that a society must solve [DEM 90], while it is also usual to assume that all agents are capable of the same language uses and interaction processes.

Only that the last assumption should not hold in general, when we deal with agent migrations between open societies, for open societies need not adhere to the same patterns of language and interactions, and migrating agents may face both language and protocol difficulties when moving from one society to another. So, a general ability to recognize social roles and social processes in the level of language and interaction is also demanded from migrating agents and, reversely, from open societies able to accept agents that come from abroad.

In the rest of the paper, we discuss the particular problem of the **identification of agent roles** in interactions performed by abstract agents living a simple, abstract model society. The model society that we consider is the producer-consumer society already study initially in [COS 93b]. Here, a third type of agent role (namely, the **intermediaries**) is also included in the society, besides the **producers** and the **consumers**. This allows for more complexity in the structures and processes involved in the problem, which serves as better a test for the functional approach.

6. The Producer-Intermediary-Consumer society

To represent protocols we use transition systems specifications [PLO 81]. A protocol action is assumed to change the states of the interacting agents (in the PIC model, only two agents are affected by each action).

Transitions have the general form **($\underline{\text{X}}$,Y) message -> (X',Y')**, where **X**, **X'**, **Y**, **Y'** are agent states, and the underline indicates the sender of the message. An agent may send a message only if it is in the state indicated in the corresponding transition of the protocol. The notation we use was also influenced by the CSP notation [HOA 85].

State-transitions not performed by actions not under the control of the protocol (such as the effective production and consumption actions, in the examples below) do not originate transtions in the protocol, but their effects (expressed as state transformations) are reflected in the states indicated in the protocol. They are indicated as sequences of state names, appended to agent identifiers.

To each protocol description one can associate, in a natural way, transition graphs representing executions (traces) of the protocol, as shown in the examples below. Note, however, that since each protocol may specify multiple processes, the traces of their executions are not traces, in fact. They are multiple traces combined in a single transition graph. In [COS 93a] this was expressed by saying that social processes are **parallel interacting processes**.

6.1 The main protocol

The Producer-Intermediary-Consumer society exists to allow the existence of the production-consumption process. The Producer-Consumer protocol that describes such process is thus the main protocol specifying the functioning of the society. Any other extant social process must exist for the benefit of that main process (or, at least, must not jeopardize it), which means that its protocol must be compatible with the main protocol.

protocol consumer-producer;
    C: Consumer;
    P: Producer;
    (<u>C.init</u>,P.wait) request(product) ->
        (C.wait,<u>P.produce.send</u>) inform(product(X)) ->
            (C.consume.init,P.wait).

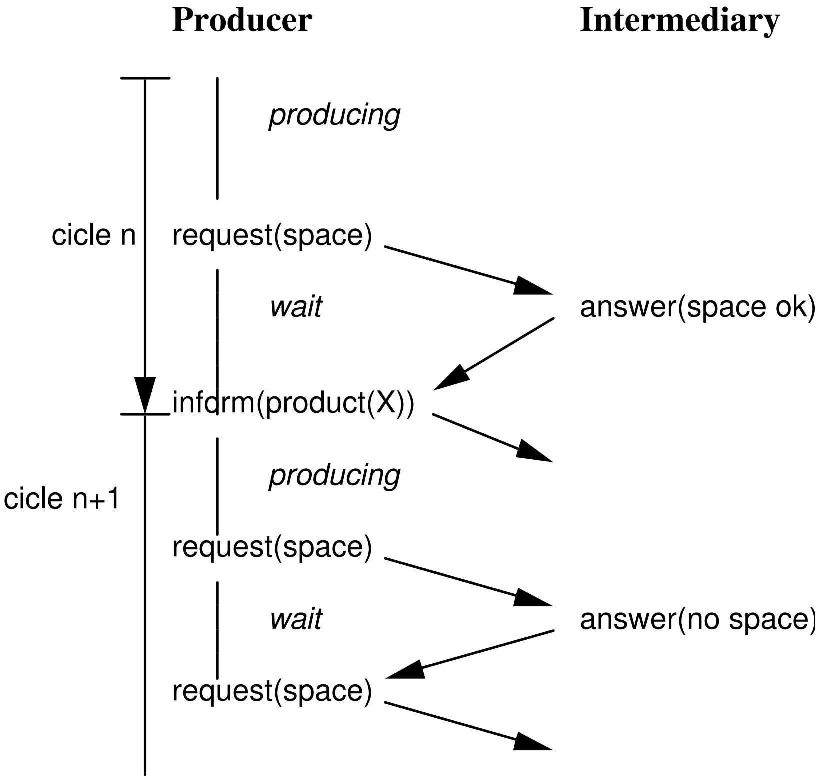
(Fig. 1: The main protocol)

6.2 The Producer-Intermediary protocol

Fig. 2 shows the Producer-Intermediary protocol. Fig. 3 shows a trace for it.

protocol producer-intermediary;
    P: Producer;
    I: Intermediary;
    (<u>P.init</u>,I.wait) request(space) ->
        (P.wait,<u>I.send</u>) answer(no space) -> (P.init,I.wait)
        or
        (P.wait,<u>I.send</u>) answer(space ok) ->
            (<u>P.produce.send</u>,I.wait) inform(product(X)) -> (P.init,I.wait).

(Fig. 2: The Producer-Consumer protocol)



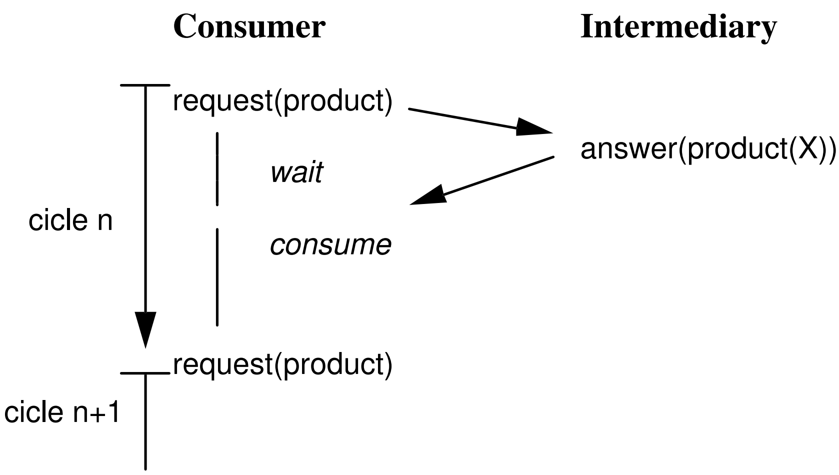
(Fig. 3: A trace of the producer-intermediary protocol)

6.3 The Intermediary-Consumer protocol

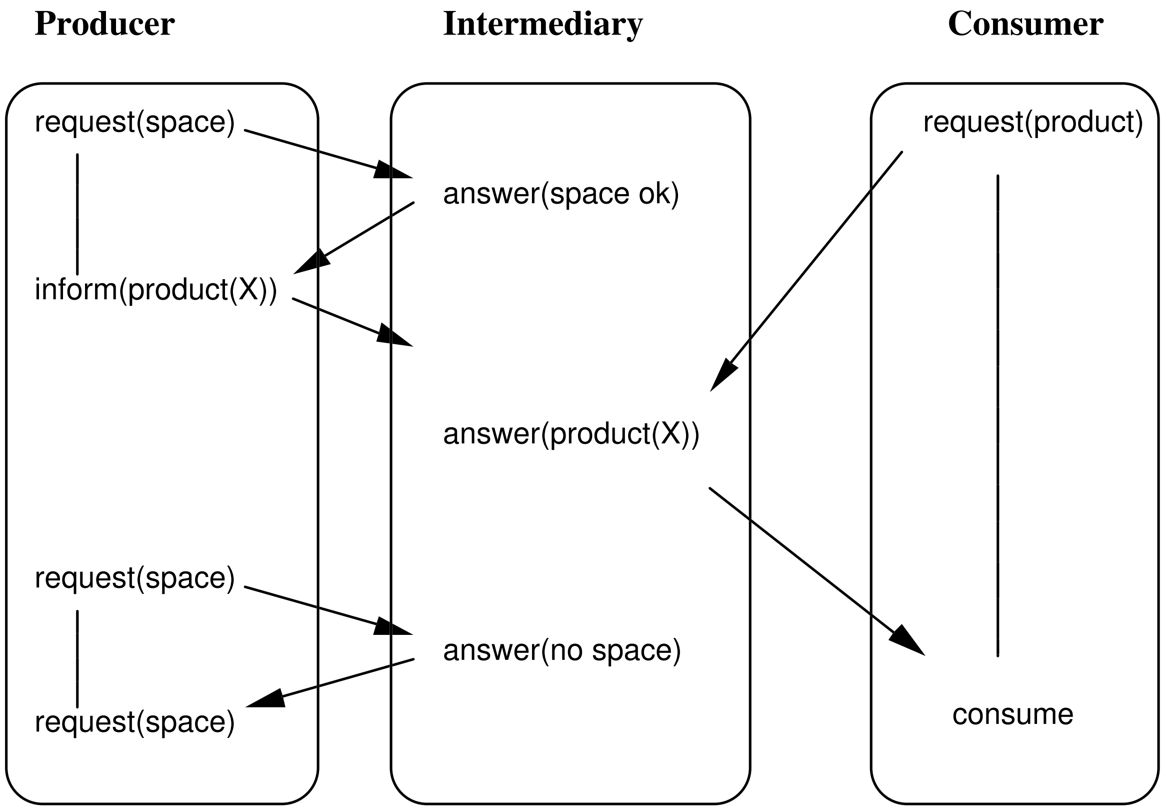
The Intermediary-Consumer protocol is shown in Fig. 4. A trace for it is in Fig. 5.

protocol consumer-intermediary;
C: Consumer;
I: Intermediary;
(<u>C.init</u>,I.wait) request(product) ->
(C.wait,<u>I.send</u>) answer(product(X)) -> (C.consume.init,I.wait).

(Fig. 4: The Intermediary-Consumer protocol)



(Fig. 5: A trace of the Intermediary-Consumer protocol)



(Fig. 6: A trace of the overall functioning of the PIC society)

6.4 The overall functioning of the PIC society

One can easily obtain an overall protocol for the PIC society. A transition-graph corresponding to such protocol is shown in Fig. 6. Note how the protocol supports the main protocol as a *sub-protocol*. Note how the basic protocols Producer-Intermediary and Intermediary-Consumer are also included in it.

Once an agent migrates into the PIC society, it must find a role for itself in the society. It can either become a new producer, intermediary or consumer, joining efforts with the native agents in the maintenance of the main social process of production-consumption, or it can find a new role for itself, if the functional specification of the society allows for the dynamical creation of social roles.

In either case, the functioning of the arriving agent must not jeopardize the functionings of the already extant agents, or at least, it must do so in a way that those a gents are able to circumvent the pertubation, maintainning the functional continuity of the society (see [COS 94] for an extensive informal analysis of the general problem of agent migration in open symbolic computing societies).

7. Mechanisms for role identification

In this section we show some mechanisms for role identification, which are being developed in [HÜB 94]. We identify three main mechanisms for role identification: explicit role presentation (suggested in [POP 93]), recognition of observed roles, and induction of role descriptions.

7.1 Explicit role presentation

The mechanism of **explicit presentation of roles** is the simplest mechanism allowing an agent to identify the role that another agent is performing in a society, in a given moment. The inquiring agent requires a role presentation, and the inquired agent replies with an information telling the role it is playing at that moment.

Note that many assumptions underlie the mechanism for explicit role presentation. First, it is assumed that an agent knows which is its current role in the society. This implies that either the notion of role is an absolute one (where the role performed by an agent can be known independently of the current functioning of the society), or the notion of role is relative to the current way of functioning of the society, and the agent is able to analyse such current functioning and establish the role it is playing in it at that moment.

In [POP 93] it is implicitly assumed that the notion of role is an absolute one, because role is identified with behavior, and it is assumed that agents know their own behaviors, and inform about them to the other agents. We take the view that role is a relative notion, meaning behavior within a social process. But to be able to introduce a simple mechanism for the explicit presentation of roles (in which agents need not investigate their own behaviors) we make the notion of role be explicitly represented in the system, with agents having the built-in information about the roles they may play in social processes.

A second assumption underlying the mechanism for explicit presentation of roles, in the case of open systems, is that the arriving agent uses the same names for denoting roles that are used by the native agents, which need not be the case if different societies follow different histories. This is the main restriction for the universal use of the mechanism for explicit role presentation.

Fig. 7 shows the protocol that controls the mechanism for the explicit presentation of roles. In the reply, **role(x)** is the role played by agent X.

```
protocol presentation;
        X: Agent;
        Y: Agent;
        (X.init,Y.wait) request(role) ->
                (X.wait,Y.send) answer(role(X)) -> (X,Y).
```

(Fig. 7: Protocol for the explicit presentation of roles)

7.2 Recognition of observed roles

Alternatively, role identification may proceed by what we call **role recognition**. Role recognition proceeds as follows. An agent, the **observer**, knows a pre-defined set of roles, the specifications of which it can access. The observer agent can use such role specifications to try to recognize the role performed by a second agent, the **observed agent**, whose behavior it is observing.

The recognition process proceeds as follows. The observer agent observes the behaviors of the observed agent and of the agents with which that agent is interacting. It uses those observations to produce a description of the behavior of the observed agent and its relations to the behaviors of the other interacting agents. This description is compared with the set of pre-defined role specifications. The comparison may tell the observer which is the role that the observed agent is playing in the social processes in which it is involved.

Naturally, the role that the observer identifies by the observation of the behavior of the observed agent, and that it recognizes as the role played by that agent at that moment, constitutes a belief about the observed agent, that the observer builds for itself. So, it is subject to revision.

The main requirements for the implementation of this mechanim are two. Firstly, that the observer agent is capable of observing the observed agent and the agents with which it is interacting, for otherwise the observer will not be able to deal with the social process in which the observed agent is involved. Secondly, there must be a formalism for the specification of roles, that the observer agent can use to produce descriptions of the observed behaviors. The formalism should also be used to represent the pre-defined set of roles which are known by the observer agent.

In [HÜB 94], the specification of a role is made through the explicit representation of all the traces of agent behaviors that the role allows. They are called **trace specifications**. The notation used is simple enough to presented through example. It is essentially the CPS notation [HOA 85] (fig. 8 presents trace specifications for the roles **Consumer** and **Producer**).

The procedure used by the observer to recognize a role, is simple. The observer follows step by step the behavior of the observed agent. While it is doing this, it is also following the possible traces of every known role. This continues until the observer agent is able to decide which of the known roles the observed agent is playing.

```
Consumer = C; Consumer.
C = message(request(product),Inter,Cons) ->
        message(answer(product(X)),Cons,Inter) ->
                consume(product(X)) -> SKIP.
alphabet(Consumer) = {
        message(request(product),Inter,Cons),
        message(answer(product(X)),Cons,Inter),
        consume(product(X))}.

Producer = P; Producer.
P = produce(X) -> P1(X) ; SKIP.
P1(X) = message(request(space),Inter,Prod) ->
```

```
                    (message(answer(space ok),Prod,Inter) ->
                            message(inform(product(X)),Inter,Prod) -> SKIP)
                    |
                    (message(answer(no space),Prod,Inter) -> P1(X))).
alphabet(Producer) = {
            message(request(space),Inter,Prod),
            message(answer(space ok),Prod,Inter),
            message(inform(product(X)),Inter,Prod),
            message(answer(no space),Prod,Inter),
            produce(product(X))}.
```

(Fig. 8: Trace specifications for roles Producer and Consumer)

In the case of the simple behaviors envolved in the PIC society, where all behaviors are cyclic, the decision about the role being performed by the observed agent occurs when the observer is able to decide that the observed agent is performing one of the cycles prescribed by one of the known roles. This is a simple minded identification procedure, suitable for the recognition of elementary roles, such as producers, consumers and intermediaries. More complex roles may need more complex recognition procedures.

7.3 Induction of role descriptions

A third type of role identification mechanism is also being studied. It is called **role induction**. The idea is that the observer agent proceeds, as in the case of role recognition, by following step-by-step the behavior of the observed agent and the agents with which it is interacting. However, instead of comparing the observed behavior with a pre-defined set of trace specifications, the observer agent analyses the observed behavior looking for typical indications that the observed behavior corresponds to a social role.

In the experiments made in [HÜB 94], for instance, the observer agent looks for **typical action sequences**. The idea is that, within certain limits, behaviors corresponding to social roles in a society can be defined as organized sets of certain action sequences that are common to every (or almost every) social role in the society.

The same language used in the role recognition process to represent trace specifications can be used to represent induced descriptions. As in the role recognition process, the induced description is a belief that the observer agent creates for itself, and which is thus subject to revision. The notation for the trace specifications can also be seen as a language where actions sequences are combined, and even a calculus of trace specifications can be elaborated, on the basis of [HOA 85], where equivalence of induced roles can be determined.

To illustrate the approach, consider the action sequence of sending messages that can be represented by: **step(X,Y) = request(X) -> inform(Y)**.

The role **producer** may then be defined roughly as:

```
producer =
        step(storespace,product) ->
                produce ->
                        producer.
```

The role **consumer** may then be defined roughly as:

```
consumer =
        step(product,storespace) ->
                consume ->
                        consumer.
```

The role **intermediary** may then be defined roughly as:

```
intermediary =
        (step(storespace,product)
        |
        step(product,storespace)) -> intermediary.
```

From the point of view of the observer agent, those role definitions may then be summarized as:

```
producer = step(storespace,product) -> producer.
consumer = step(product,storespace) -> consumer.
intermediary =
        (step(storespace,product)
        |
        step(product,storespace)) -> intermediary.
```

It is enough for the observer agent to observe enough **step** action sequences of certain forms, to create a belief that the observed agent is perfoming one of those roles.

An even more free-wheeling approach to role induction can be adopted. The observer agent can restrict itself to register observed action sequences. Whenever a set of closed cycles has been observed in those action sequences, so that the observed agent keeps behaving according to such cycles, the observed set of closed cycles may considered as the induced description of the observed role. After that, a characterization for it may be look for. For instance, the observer agent may classify it within its set of known roles, defining it as a new role, in the case of failure of the classification.

8. Conclusion

We have touched in this paper, some problems concerning agent migration between open societies. Theoretical aspects of open societies and agent migration, which are currently under investigation [COS 93a] [COS 94] have been summarized in the first part of the paper. The idea of two functional levels in an open society, the language and interaction level, and the knowledge and performance level, was introduced. They are assumed to be functional structures in the sense of [COS 94]. It was proposed that some problems concerning their specific ways of functioning could be solved by applying solutions found for corresponding problems in the general structure.

The problem of role recognition is one of such problems. The Producer-Intermediary-Consumer society, already taken as object of study in [COS 93b], was taken as the sample society where to study solutions for the role recognition problem in a general functional structure. Architectural and operational problems concerning role recognition in the PIC society are under investigation [HÜB 94], and have been summarized in the second part of the paper. Prototype PIC societies have been implemented on the workbench developed by Moniz [MON 93]. They have served for the ellaboration of the role recognition mechanisms. Explicit role presentation and role recognition have found somewhat direct realizations. Role induction is currently being examined.

Language support for agent migration between open societies is also being investigated [BOR 94]. The work aims at determining general language mechanisms which relieve agents that are migrating to a new society from knowing, previously, language and interaction features specific to that society (in particular, protocols for establishing elementary interactions).The migrating agent should then be able to participate in simple protocols in the new society, without having to pass through complicated translation sessions which could amount to its complete reestructuring, from the point of view of language and interaction.

9. Bibliography

[BER 92] BERTHET, Sabine, DEMAZEAU, Yves, BOISSIER, Oliver. Knowing Each Other Better. In: *11st International Workshop on Distributed Artificial Intelligence*. Glen Arbor, 1992. p. 1-20.

[BOR 94] BORDINI, Rafael Heitor. Suporte linguístico para migração de agentes. CPGCC/UFRGS, 1994. (M.Sc.diss., to appear).

[COS93a] COSTA, A. C. Rocha, CASTILHO, J. M. V., CLAUDIO, D. M. *Towards a constructive notion of functionality*. II-UFRGS, 1993. (to appear).

[COS93b] COSTA, Antônio Carlos da Rocha, CASTILHO, José Mauro Volkmer de, CLÁUDIO, Dalcidio Moraes. Functional Processes and Functional Roles in Societies of Computing Agents. In: X *Simpósio Brasileiro de Inteligência Artificial.* Porto Alegre: SBC,1993. p. 267ss.

[COS 94] COSTA, Antônio Carlos da Rocha. *Open Physical Symbol Systems*. II-UFRGS, 1994. (to appear).

[DEM 90] DEMAZEAU, Yves, MÜLLER, Jean Pierre. Decentralized Artificial Intelligence. In: *Decentralized Artificial Intelligence1*. Organização: DEMAZEAU, Yves, MÜLLER Jean-Pierre. North-Holland, Elsevier Science Publishers, 1990. p. 3ss.

[HOA 85] HOARE, Charles Antony Richard. *Communicating Sequencial Processes*. UK: Prentice-Hall, 1985.

[HÜB 94] HÜBNER, Jomi Fred. Estudo sobre entrada de agentes em sistemas multiagentes abertos. CPGCC/UFRGS, 1994. (M.Sc. diss., to appear).

[MON 93] MONIZ, Luis Manuel Ferreira Fernandes. SSAA: Sistema para Simulação de Agentes e Ambientes. Universidade Técnica de Lisboa, junho, 1993. Dissertação de Mestrado.

[PLO 81] PLOTKIN, Gordon. A Structural approach to Operational Semantics. Comp. Sci. Dept., Aarhus Univ., 1981.(DAIMI FN-19, Sept. 1981).

[POP 93] POPULAIRE, Philippe et al. Description et Implementation de Protocoles de Communication en Univers Multi-Agents. *IèresJournées Francophones sur I'Intelligence Artificielle Distribuée et les Systèmes Multi-Agents*, Toulouse, AFCET & AFIA, 1993.