Journal of Achievements in Materials and Manufacturing Engineering

VOLUME 55
ISSUE 1
November
2012

Co-ordination in the autonomous software agents' systems

J. Madejski*
Institute of Engineering Materials and Biomaterials, Silesian University of Technology, ul. Konarskiego 18a, 44-100 Gliwice, Poland
* Corresponding e-mail address: janusz.madejski@polsl.pl

Received 21.09.2012; published in revised form 01.11.2012

Analysis and modelling

ABSTRACT

Purpose: Agents are designed to behave individually rational, which means that they should maximize their personal utility which is the way to make them less vulnerable to mean actions of others, yet they have to co-ordinate their actions to reach common goals, which is the purpose of this work.
Design/methodology/approach: Agents can create and pursue their individual goals, behaving in a 'selfish' way to acquire the desired state of their world. To achieve that they may choose to adopt goals of other agents too, should this co-ordination be assessed as beneficial for them. Moreover, there is also a possibility to define the desired states of the agents in a way which will induce them to work together rather than try to operate individually. This may include their specialization, which forces in most cases sharing of their potential. This may be achieved by specialised design of agents being able to carry out elementary tasks. Such approach calls however, for design of a layer of supervisory agents which will be capable of realising what is the multi-agent overall system goal, setting up their teams from simple agents and committing to common sub-goals. All such systems may be efficiently developed only after careful study of the successfully operating systems in which humans are the agents, whose tasks may now be assigned to the software ones. These agents have to be coupled, as it also happens in their human counterparts.
Findings: Development of the software agents' co-operation framework based on review of publications covering both the fundamental considerations, as well as the latest developments.
Research limitations/implications: Approach presented still needs careful testing and refinement of the co-ordination / negotiation rules.
Originality/value: Co-ordination of agents to reach their common goal, satisfying also their individual utility.
Keywords: Artificial agents; Virtual enterprise; Scheduling; Negotiation; Co-ordination in multi-agent systems

Reference to this paper should be given in the following way:
J. Madejski, Co-ordination in the autonomous software agents' systems, Journal of Achievements in Materials and Manufacturing Engineering 55/1 (2012) 74-79.

1. Introduction

Efficient and timely collection and access to data describing the manufacturing system status feature an important issue in development of the decision making systems that will perform properly their tasks in the dynamic environment. Development of the relevant systems calls for compiling the experience gathered over the years in the system served by human 'agents'. Such knowledge base may be later used first to mimic the behaviour of the system controlled by human operators, and later - to populate it, sometimes modified, 'cleaned' and optimised as the decision

making system prototypes for new manufacturing systems. The fuzzy models based on this knowledge are event oriented and represent single agents which - when needed - should be able to solve jointly problems exceeding the capacity of a single one. To this end negotiation skills are needed which lead either to delegation of a task to a single agent or in setting up an ad-hoc task group to handle the problem. This approach makes it possible to model the required co-ordination framework needed to set up, carry out, and finalise the negotiations focused on attaining the individual goals, adding to reaching the overall system's goal [1].
Co-ordination is always needed when many agents are needed to form a task group. Forming a group requires always exchange of information between agents, which - in turn - calls for establishing the common protocols to make such communication possible. Some protocols are discussed and modifications are suggested to make them more suited to specific application domains [2-5].

The notion of the Blackboard is presented which lets the agents communicate freely, without limiting their contacts to bi-lateral ones only. This way the particular agents may either commit to carry out the new task, provided they find it beneficial for their own utility assessment, or look for assistance, or simply get involved in negotiations with others, coordinating their tasks. Meeting this requirement within the restricted time-frame calls for the layered system architecture, as reaching a solution in real time does not allow lengthy deliberation process [6-8].

2. Modeling of Multi-Agent Systems

The notion of Multi-Agent Systems (MAS) and their proposed implementations become more and more complex and can take over many manufacturing control process and execution tasks from the traditional systems in which humans are responsible for the smooth flow production flow. An important issue is modelling of their flexibility and interaction types which are the base for their organization models [9]. The main feature of the MAS systems is their distributed nature, which makes modelling of the agents from which they are composed and their interactions a complex task. Modelling requires specifying the environment in which the agents operate, next, their nature has to be defined, and finally their co-operation modes, which is always based on a number of interactions between the particular agents. Moreover, these interaction mechanisms have to be designed in a way which will make the entire MAS system - as a whole - capable to fulfil the given task, and prevent the unwanted disagreements, or selfish behaviour or 'negligence' of the agents expected to collaborate. There are a number of methodologies used to attain this goal, e.g. PACO [10] taking a simple approach of the agents. Anyway, the main goal is to implement a model of collaborating the reactive agents acting in a certain environment, where all agents may have different nature and which partial solutions of a the global problem posed to the system [13-14].

The efficient and possibly uncomplicated agent paradigm states that that they are wholly reactive, devoid of their an continuously updated internal representation of themselves, other agents, or the environment. Therefore, they have to respond necessarily to all changes of the environment. The notion of the environment is their perception of their world which they perceive by interaction realised by getting and sending messages. This way, events make them react appropriately, regardless if an even would be an expected or unexpected one. Good examples might be: arrival of a new part or batch of parts for processing, breakdown of some technological equipment, planned maintenance, etc. In all these cases the agents constituting the MAS have to search for a new solutions through their interactions, which may be described as the equilibrium state in the PACO approach mentioned earlier as an example [15-18].The main problem in modelling the Multi-Agent Systems is defining the nature of the agents from which they are composed. The simple, yet effective strategy is to specify the agent components, which , may include, for instance the following components [19-21]:

- *perception field:* specifying what the agent can learn about its environment,
- *communication field:* defining the list of agents with which the particular agent may interact with,
- *action field:* laying out the space for an agent where it mayact.

The agent's environment is the space in which it exists, moves, and interacts with others. It should be noted that the space nature may be of various types, like the informational or conceptual. In most manufacturing systems, however, the agent's environment is simply a model of the physical space in which the MAS agents reside and operate.

The environment is more than just a space being also a resource, whose status, at a particular time, can be acquired by the agents. The environment's state is advised by its controller monitoring its current status and maintaining its occupation plan - as a list of time slots allocated to certain agents, or being still free to use. Being organised is beneficial to agents thanks to their objective function rewarding them for successful completion of tasks. Such organisation may result either from the MAS design, modelled features of the agents, or from direct interactions taking place ad hoc between agents aimed at completion of a task. Interactions are required to advise the tasks, find collaborators and reserve environment time slots. The simplest way to model the interactions is to is to define them as speech interactions with the specific protocol, however, there are also approaches which treat the interactions as forces, being spring-like or electrostatic in nature [10].

2.1. Model of an Agent

Let us consider the materials processing as the example of agent implementation domain (Fig. 1). Each process plan consists of a number of operations. Each operation has to be completed using the relevant resource (we mention one resource at a time, to make the example simple). The resources, as elements of the environment may be contacted via their controllers. These controllers, can advise - when approached - what time slots are available, moreover, they can monitor condition of the important resource parameters. As an example, the resource like the heat treatment oven may be characterised by such parameters like temperature, atmosphere, chamber dimensions, etc.

A number of agents are generated for each process plan - one for each operation. Each agent is provided with the necessary knowledge referring to the operation, i.e., its parameters, like e.g., resource type, hardening temperature and time, and with information about the required resource and its immediate neighbouring agents - preceding one and the successive one. No global information is provided about the process as such, i.e., other agents representing other operations or other process plans.

As shown in Fig. 1, process plans, consisting of operations which have to be carried out at a predefined sequence, are treated as groups of agents which complete the product by acting one by one. Clearly, these agents act in a certain environment - requiring various resources. In our example, to make the case simple, every agent needs some resource (for clarity only some links are shown in the figure above), which means that there are queues of agents waiting for their resource.

There may be several ways in which such queues are managed - from the simplest and obvious static FIFO strategy to dynamic ones with priority rules. Moreover, the nature of some resources may allow several agents to use them at the same time slot, provided the environment status (e.g., furnace temperature, treatment time, etc.) may be the same for all of them, provided there is enough space for them (decided by, e.g., chamber size).

All agents allocated to certain resources form groups which, albeit different in nature, have always the same goals:

- Proceed to the relevant resource,
- Stay close to the predecessor agent of its group (keeping an order in the queue and minimising delays).

The goals above guarantee that the agent will go to the right resource and will eagerly make use of it. However, this approach alone does not ensure any co-operation between the agents, therefore, some constraints have to be added [22-27], e.g.:

- Assist the succeeding agent from the group to stay close (thus inviting, if possible, to use the same time slot on a resource),
- Help other agents that might use the same time slot to fulfil their goal (this may boil down to spatial rearranging the agents to make room for yet another one).

When agents from two groups would like to use the same time slot for a given resource they have to be able to negotiate who will win it.

2.2. Co-ordination of Agents' activities

According to [1-3, 5, 8], an agent is characteristic of a certain internal state, which is unavailable to other agents, unless the agent chooses to let them know about it, and can decide what to do next according to this state. As mentioned above, the agent's architecture makes it possible to:

- Interact with other agents,
- Make decisions based on its internal state and knowledge,
- Be conscious of its internal state, having a memory.

An agent can act and interact - communicate - using an interface for reception and sending messages. The received messages are saved on a stack, so that the agent can process them in due course, as it may be busy with another one when other messages arrive. Interaction of agents consists in the following phases (Fig. 2):

- Message reception,
- Message processing,
- Relevant action.

The agents send messages as their actions, which may be carried out as agent's initiative without external stimulation, coming from its domain knowledge and previous actions. Now we need to specify what are the characteristic features of the environment in which the agent operates. The features set forth below make interaction planning possible - and predictable. The environment, referred to also as resources, are characteristic of being:

- Static, which means that it does not change if not acted upon by an agent,
- Discrete, i.e., stable with a finite number of states,
- Deterministic - future states result from its current state and actions of an agent,
- Attainable, so than an agent can obtain information about the state of the environment.

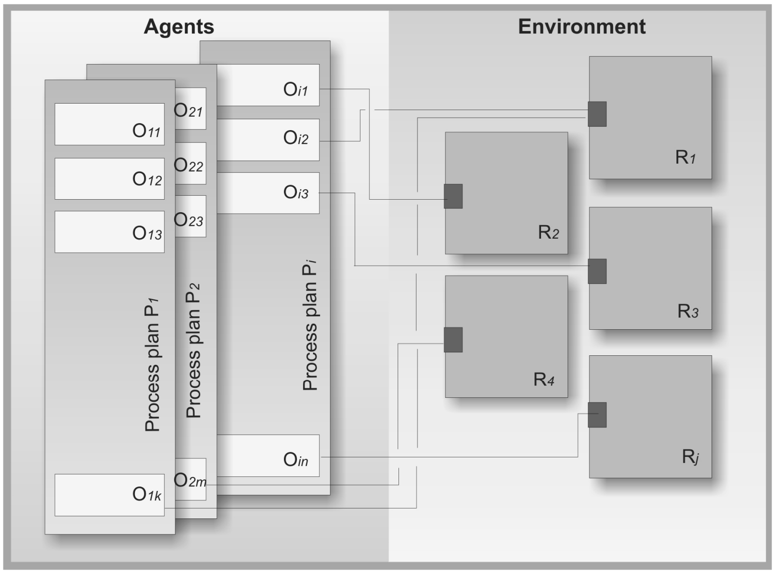

Fig. 1. Process plans as sets of agents **O** operating in the environment composed of resources **R**

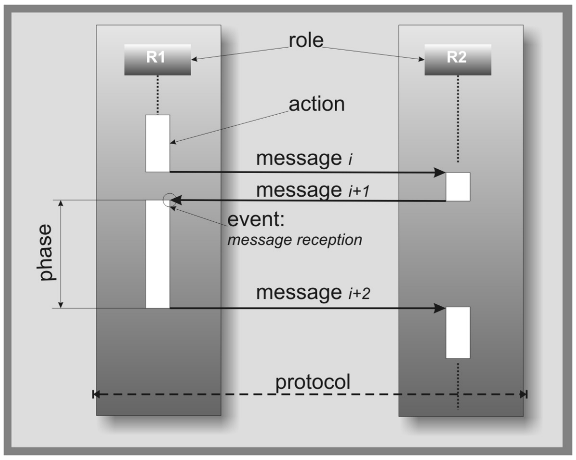

Fig. 2. Illustration of the agent interaction protocol

The interaction between agents consists in sending and receiving messages between agents. The interaction is performed by sending one message among the set of messages that sending agent is able to send and/or in the reception of a message among the messages that the agent understands. There may also be messages that are incomprehensible to the receiving agent.

The interaction protocol is a sequence of messages exchanged between agents playing at least two roles [28].The interactions, in their simplest form, may be defined between two agents at a time only, without referring to any other agents, or without any knowledge of the entire system. The resources occupy certain locations in the system space, so the agents, knowing their current locations may assess the time when they will arrive at the particular resource's queue. Any co-ordination between the agents requires some form of interaction (also with the resource controller, which is also treated as an agent).

When agents interact, e.g., by means of an ACL [29], the meaning of such exchange is characterized by communicative acts which belong to one of the following categories - Table 1.

Any interaction between the agents has to be commenced and completed in an orderly way as they play their roles - collections of phases governed by events [10,19,30]. A conversation moves from one state to another, according to the given state transition diagram. It may be either a one-time communication act (ask-one), or a continued one (subscribe). Figure 3 illustrates the exemplary agents' ask-one conversation policy for Jackal [30] described using the Deterministic Finite Automata (DFA) model. In such model, each conversation starts with a state called START, and ends with a state called STOP, which guarantees that the interaction can eventually come to an end.

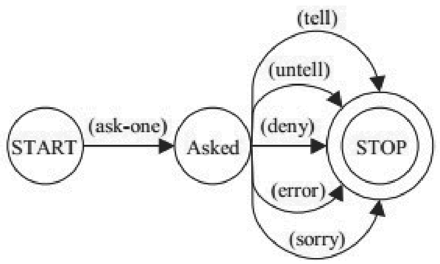

Fig. 3. Example of the DFA's for ask-one conversation [30]

Table 1.
Categories of communicative tasks

Category	Description
representatives	represent the state of affairs, e.g., the furnace chamber temperature, flow rate of the shield gas, etc.
directives	compel the receiving agent to carry out some task or respond to an enquiry, e.g., `"Put the workpiece into the quenching bath."` or `"What is the sample hardness after hardening?"`
commissives	induce the receiving agent to commit itself to do something, e.g., to declare `"I will carry the charging tray to furnace No 3"`
expressives	advising impressions representing the agent's psychological state; differs from the Representatives by advising the agent's state, not providing statements about other entities, e.g., may provide assessments based on quality control results, like `"The batch should be rejected due to surface defects"`.
declaratives	stating the decisions made, or tasks completed, e.g., `"Batch ID is J-23"`
permissives	giving permission for action which may be suspended until a decision is made, e.g., `"Ship the goods as per Order ID 2012Exp"`
prohibitives	statements banning some actions, e.g., `"Do not switch conveyor on."`

2.3. Postprocessing of messages

The following notations are used to explain the interaction process:

- M^{An} is the set of messages that may be sent by agent *An*;
- $m^{An}{}_{Sent,\ t}$ is the message sent by agent *An* at the moment *t*;
- $m^{An}{}_{Received,\ t}$ denotes the message received by agent An at the moment t;
- m^{An} message that incomprehensible to agent A_n.

Therefore the interaction between agents A and B may be defined as:

$$Interaction_{An,\ Am} : M^{An}_{Sent} \rightarrow M^{Am}_{Received} \cup \{m^{Am}_{\varphi}\}$$

Processing the message and triggering the eventual action is done in two successive steps:

- *memorization* represented by the change at the internal agent's state caused by the received message,
- *decision* selection of the relevant action (there may be a case in which the messages was understood, however no pertinent action is defined).

We can represent this as follows:

S^{An}	- set of possible internal states of agent A_n;
$s^{A}{}_{i}$	- internal state of agent A_n at moment *t*;
AC^{An}	- set of actions that may be taken by agent A_n;

${ac^{An}}_i$ - action taken by agent A_n at moment t;
ac^{An} - denotes that no action is taken by agent A_n;
therefore:

$$Memorization_{An} : M^{An}_{\mathrm{Re}\,ceived} \times S_{An} \rightarrow S_{An}$$

$$Decision_{An} : M^{An}_{\mathrm{Re}\,ceived} \times S^{An} \rightarrow AC^{An} \cup \{ac^{An}_{\varphi}\}$$

Decisions triggered by interactions are defined by a set of rules specified by the system designer. The exemplary rules may have the form as shown below:

Rule 1 IF there is a possibility to increase the time slot size for the action
THEN

$$T_{now} = T_{min} + K\,(T_{max} - T_{min})$$

where:
- T_{now}, T_{min}, and T_{max} are the current, minimum, and maximum time slots;
- K is a static constant.

Rule 2 IF another agent, intending to use the same resource, comes closer than a given distance, THEN reduce the current time slot:

$$T_{now} = T_{min} + M$$

where:
- T_{now}, is the current time slot;
- M is a static constant - margin.

Rule 3 IF the succeeding agent is unable to stay close enough to its predecessor
THEN increase the current time slot:

$$T_{now} = T_{max}$$

Sometimes the agents may block themselves in their strive to use their resources, so the eventual collisions have to be resolved by either jumping over another agent's time slot, pushing it to a later time, or switching places with a blocking agent (A_n).

Rule 4: IF $T_{now\ Ai} > T_{now\ Ai+1}$ and a time slot of at least T_{min} is available between the end of agent A_{i+1}'s time slot and the length of $T_{now\ Ai}$ time slot
THEN agent A_i goes to the other side of A_{i+1}
where:
- $T_{now\ Ai}$, $T_{nowAi+1}$ - time slots of agents A_i and A_{i+1} respectively

Rule 5: IF there is no time for agent A_i beyond agent A_{i+1}, yet A_{i+1} wants to retract
AND
IF $T_{now\ Ai} > 0.5\ T_{min\ Ai+1}$
THEN they take each other's positions.

Rule 6: IF neither Rule 4 and Rule 5 fired
AND
IF A_i still insists on acquiring a part or entire $T_{now\ Ai+1}$ time slot

THEN A_i begins to negotiate taking into account the objective function value of A_i and A_{i+1}

IF A_i wins, then A_{i+1} is pushed away
ELSE they stay where they are.

Decisions made by the agents result in actions taken and may trigger a locutionary act of one of the communication types as specified in Table 1 above.

3. Conclusions

The interactions in Multi-Agent Systems aimed at co-ordination of agents for execution of tasks may be designed at two levels of abstraction: micro- and macro levels. The micro level, specifying relations between the agents, is presented in this paper. The next stage of research should be focused on defining both the negotiations among the agents at the same layer (horizontal) and also the interlayer interactions (vertical) to be carried out by the supervisory agents [20]. This approach will make possible to design agents that will be able to be allocated many tasks at a time, and will be able to carry out the many-to-many negotiations.

References

[1] J. Madejski, Survey of the agent-based approach to intelligent manufacturing, Journal of Achievements in Materials and Manufacturing Engineering 21/1 (2007) 67-70.

[2] J. Madejski, Agents as building blocks of responsibility-based manufacturing systems, Elsevier, 2000.

[3] J. Reaidy, P. Massotte, D. Diep, Comparison of negotiation protocols in dynamic agent-based manufacturing systems, Elsevier, International Journal of Production Economics 99 (2006) 117-130.

[4] T.N. Wong, C.W. Leung, K.L. Mak, R.Y.K. Fung, Dynamic shopfloor scheduling in multi-agent manufacturing systems, Expert Systems with Applications 31/3 (2006) 486-494.

[5] G. Gaspar, Communication and belief change in a society of agents, Towards a formal model of autonomous agent, Y. Demazeau, J.-P. Müller (Eds), Decentralized AI II, Elsevier Science Publishers B.V., Amsterdam, 1991.

[6] J. Campbell, M. d'Inverno, Knowledge interchange protocol, Y. Damazeau, J.-P. Müller (Eds), Decentralized AI I, Elsevier Science Publishers B.V., Amsterdam, 1990, 63-80.

[7] B. Burmeister, A. Haddadi, K. Sundermeyer, Generic, configurable, cooperation protocols for multi-agent systems, C. Castelfranchi, J.-P. Müller (Eds), From reaction to cognition 957, Springer Verlag, Berlin, 1995, 157-171.

[8] J. Madejski, Modelling of the manufacturing system objects interactions, Journal of Achievements in Materials and Manufacturing Engineering 24/2 (2007) 167-170.

[9] S. Sian, Adaptation based on cooperative learning in multi-agent systems, Y. Demazeau, J.-P. Mülller (Eds), Decentralized AI II, Elsevier Science Publishers B.V., Amsterdam, 1991.

[10] J.W. Perram, Y. Damazeau, A multi-agent architecture for distributed constrained optimization and control, Proceedings of the Scandinavian Conference on Artificial Intelligence SCAI'97, G. Grahme (Ed.), IOS Press, 1997, 162-175.

[11] R.A. Mahdavinejad, A new approach to job shop-scheduling problem, Journal of Achievements in Materials and Manufacturing Engineering 41 (2010) 200-206.

[12] J. Madejski, Dynamic scheduling for agent based manufacturing systems, Journal of Achievements in Materials and Manufacturing Engineering 40/1 (2010) 66-69.

[13] S. Toshiharu, F. Kensuke, H. Toshio, K. Satoshi, Effect of Alternative Distributed Task Allocation Strategy Based on Local Observations in Contract Net Protocol, in N. Desai, A. Liu, M. Winikoff (Eds.): Principles and Practice of Multi-Agent Systems, Proceedings of the 13th International Conference PRIMA'2010, Kolkata, 2010, Revised Selected Papers, Springer, 2012.

[14] D. Wang, S.V. Nagalingam, G.C.I. Lin, Development of an agent-based Virtual CIM architecture for small to medium manufacturers, Robotics and Computer-Integrated Manufacturing 23/1 (2007) 1-16.

[15] M.D. Reimann, J. Sarkis, An intelligent system for automating the inspection of manufacturing parts, Design and Implementation of Intelligent Manufacturing systems, from expert systems, Neural; Networks, to Fuzzy Logic, H.R. Parsaei, M. Jamshidi (Eds), Prentice Hall PTR, 1995, 19-38.

[16] F.P.M. Biemans, Manufacturing Planning and Control, A Reference Model, Elsevier, 1990.

[17] L.R. Nyman, Making manufacturing cells work, Society of Manufacturing Engineers in Cooperation with the Computer and Automated Systems, 1992.

[18] S. Ossowski, Co-ordination in artificial agent societies, Social Structure and Its Implications for Autonomous Problem-Solving Agents, Springer, 1999.

[19] J.P. Müller, The design of intelligent agents, A layered approach, Springer, 1996.

[20] J. Madejski, The autonomous agent-based manufacturing systems architecture, Proceedings of the Scientific Conference on the Occassion of the 55th Anniversary of the Faculty of Mechanical Engineering of the Silesian University of Technology in Gliwice M2E'2000, Gliwice, 2000, 293-302.

[21] J. Madejski, Fuzzy logic approach to the autonomous agent task utility function evaluation, Proceedings of the 8th International Conference on "Achievements in Mechanical and Materials Engineering" AMME'99, Rydzyna, 1999.

[22] H. Van Brussel, L. Bongaerts, J. Wyns, P. Valckenaers, T. Van Ginderachter, A conceptual framework for holonic manufacturing: Identification of manufacturing holons, Journal of Manufacturing Systems, Journal of Manufacturing Systems, 1999, http://findarticles.com/p/articles/mi_qa3685/is_199901/ai_n8841020/pg_1.

[23] A. Dobrzańska-Danikiewicz, D. Krenczyk, The selection of the production route in the assembly system, Journal of Achievements in Materials and Manufacturing Engineering 17 (2006) 417-420.

[24] H. Karun, P. Valckenaers, M. Kollingbaum, H. Van Brussel, Multi-agent coordination and control using stigmergy, Computers in Industry 53/1 (2004) 75-96.

[25] P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos, A. Perini, Tropos: An agent-oriented software development methodology, Journal of Autonomous Agents and Multi-Agent Systems, 2004.

[26] L. Cernuzzi, G. Rossi, On the evaluation of agent oriented modeling methods, Agent Oriented Methodology Workshop, 2002.

[27] H. Bincheng, L. Jiming, J. Xiaolong, From local behaviors to global performance in a Multi-Agent System, Intelligent Agent Technology, 2004.

[28] Foundation for Intelligent Physical Agents: http://www.fipa.org.

[29] Janus - multi-agent platform, http://www.janus-project.org/Home, 2012.

[30] Y. Peng, T. Finin, B. Chu, Y. Labrou, R.S. Cost, B. Chu, J. Long, W. Tolone, A. Boughannam, An agent-based approach for manufacturing integration, The CIIMPLEX Experience,http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.56.401.