FORMAL SPECIFICATION USING AGENTS CONCEPTUALIZATION

Michel Coriat
Laboratoire MASI, Institut Blaise Pascal / CNRS - UA 818
Université de PARIS VI, 4 place Jussieu 75252 Paris cedex 05, France
e-mail: coriat@masi.ibp.fr, Fax: 33 1 44 27 62 86

Abstract

This paper tries to bridge the gap from agent conceptualization to formal specification in the field of requirement engineering. The IAM (Intentionnal Agent Method) approach aims to integrate the agent ontology with Time Petri Nets (TPN) in the development process helps in the construction of efficient and safe large software system.

TPNs deals formally with concurrency. Mathematically, it allows formal verification of safety (what a program is allowed to do), liveness (what a program must do) or quantitative properties (performance, response time, message delivering,...).

Keywords: agent, interaction, intention, formal specification, safety, liveness, Petri nets

1. INTRODUCTION

Along information systems specification, the requirement engineer must analyze and understand the problem to be solved, to provide a complete and non-ambiguous requirement document.

The difficulty is to support a natural mapping of the customer wishes, ensuring expressiveness and structuring. Because formal languages are based on mathematical concepts, they provide formal verification and formal refinement (through proof obligations) into executable code. However, structuring and expressiveness are merely provided by mathematical models. Therefore, this paper proposes a formal approach called IAM (Intentional Agent Machine). IAM integrates the agent [Bond 88, Shoam 93] ontology of concept, whose dynamic is formally expressed by means of Petri nets [Jensen 91].

Petri net theory [Jensen 92, Murata 89] unifies data and behavior in a single model, and analyze information processing at the requirement level. Tokens are used to model concurrent activities of systems. Colored Petri net (CPN) [Jensen 91] provides in particular the notion of colour to structure important nets.

To reason about agent's actions, the agent is analyzed (i) at the interaction level through tasks decomposition and allocation among the multi-agents system, (ii) considering agent's intentions which guide the behavior. The integration and study of some non-functional requirements concerning the system such as safety (what a program is allowed to do), liveness (what a program must do) and quantitative properties (performance, response time, ...) are done through Petri nets formal model. An earlier integration of these considerations in the development process leads to the construction of efficient and safe large software.

The paper is organized as follows. In section 2, we talk about the multi-agents problematic for requirement engineering and specification. In section 3, the IAM method is developed. Section 4 concludes the approach and presents future works.

2. INFORMATION SYSTEMS INTERESTS

An useful elicitation divides the domain knowledge conceptually into two parts: the static (refers to the 'what') and strategic knowledge (describes problem solving activities of 'how' to perform the tasks using the 'what').

Studying the 'what' and 'how' is no sufficient because they reflect the static, and makes verification related to the dynamics not feasible. The dynamics may be studied considering the 'when' and 'why' views [Yu 94]. The 'when' refers to actions and related state change. To better understand and abstract system actions along the time, the 'why' statement gives an integrated view of actions and state changes.

Nevertheless, the ultimate aim is to get an integrated view of the model to elicitate. The 'what', 'why' and 'how' are hence studied using the agent conceptualization and the 'when' with Petri nets.

2.1. Multi-agents system

Rather than viewing the agent with causal actions and ordinary attributes, it is characterized with *intentional* notions [Dennett 87, Cohen 90, Wooldridge 94] which allows abstraction of causality of agent's actions behind a why statement. Relationships between intentions define the agent's dynamics, i.e. causal dependencies with other agents.

2.2. Proofs

To provide quality software, hybrid methods, like Z or B [Abrial 93], provides a proof step called 'proof obligation': model the system and add invariants which must always be verified on that model. Basic verification techniques concerns completeness (i.e. required functionalities have to match offered ones) and contradiction (i.e. a specification that can be worked out from different viewpoints). Other verification techniques concerns non-functionnal requirements related to properties of distributed systems: safety (for example deadlock or mutual exclusion), Liveness (fairness or request honesty) and quantitative (response time, guaranty to deliver a message)

3. THE IAM APPROACH

The IAM approach (figure 1) divides the specification process into three steps: view, agent and proof.

At the view layer, the analyst gives an interaction graph relating dependencies between agents in terms of tasks assignment. The analysis of agents dialogs evolving into interactions is established with a speech acts dictionary relating all speech acts.

At the agent layer, the analyst describes agent dynamic and enumerates related intentions such as belief, desire, commitment and capability. Agent intentions are then related to affect each other. Afterwards, agents' (speech) acts are linked.

At the proof level, non-functional invariants are stated on agent's intentions. These invariants are to be proved on the agent model using a modular Petri nets based verification technique.

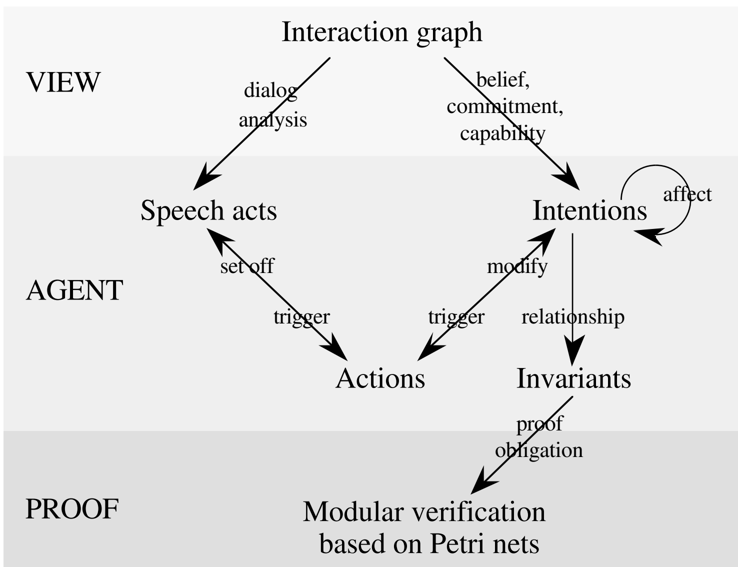


Figure 1 - overview of the analysis steps of IAM

3.1. Specifying static

The activity of an agent system refers to its coordination. Activities (tasks) decomposition and allocation are dealt with by interactions. *Activities decomposition and allocation* is done by goals decomposition, assigning resulting tasks to agents. There exist several kind of tasks assignment:

- *Organization* : framework of coordination where each agent has a specific task or a role [Wirfs-Brock 90] to do and a shared long-term goal),

- *Result-sharing*: tasks are fully predictable and static and changes are signaled to agents),
- *Planning*: activity consist of an explicit division and synchronization of tasks among agents with common goals
- *Task-sharing*: concerns tasks which can not be identified a priori. These tasks are allocated by finding an appropriate agent to execute it.

Once interactions are defined, the agent (figure 3) is defined by its behavior (subject to internal and temporal events), mental attributes, and contributions (subject to external events). Attributes are characterize agent's intentions and contributions by speech acts analysis.

3.1.1. The interaction graph

The interaction graph is a hierarchical entity-relationship like graph relating interactions among agents. Interactions are represented by dependencies in terms of task assignment.

3.1.2. Defining intentions

Once an agent's interactions are defined, it is possible to state attribute through intentions analysis: belief is defined as a (true or false) local agent knowledge. Capability states preconditions on agent's resources, by enabling the agent to do actions at precise time. Commitments define agent's goals carried out for committed agents (including itself). Because task decomposition and allocation define goals, commitment are derived from interactions. Commitment emerges from a given goal defined in an interaction. Desire expresses the agent's waiting for commitments or action. When an agent wants to commit or act it may before need an unavailable belief or a capability and then it waits for them before doing any commitment.

3.1.3. Refining interactions through speech acts analysis

The*content of the interaction* is a dialogue analysed through speech act theory [Austin 62,Holm 94, Nellborn 94]. We have retained basic speech acts such as inform, request, bid, abstain and cancel. This dictionary can be enriched by adding domain-dependent speech acts. The analyst must take care not to express redundant speech act depending of the domain context. For example, the speech act 'order' is the same as 'request' if it concerns an utterance from employer to an employee.

3.1.4. Invariants

Invariants can be defined on state or on operations. Invariants relating to safety can be expressed with first order predicate logic. But, invariants expressing liveness must take into account contextual meanings, therefore, they can not be expressed with first order predicate logic, instead they need temporal logic [Pnueli 92]. Logic formulas are expressed on intentions components (belief, desire, capability, commitment) of the agent.

3.2. Agent structure

The agent abstract machine is described by its behavior, a set of exchange (operations) and invariants definition (see figure 2). An event occurs when a change happens in the agent abstract machine .

Behavior is described by interactions and agent's intentions described with Petri nets. Invariants are defined by temporal logic formula on agent intentions. Exchange and interaction structures are described in section 3.2.2.

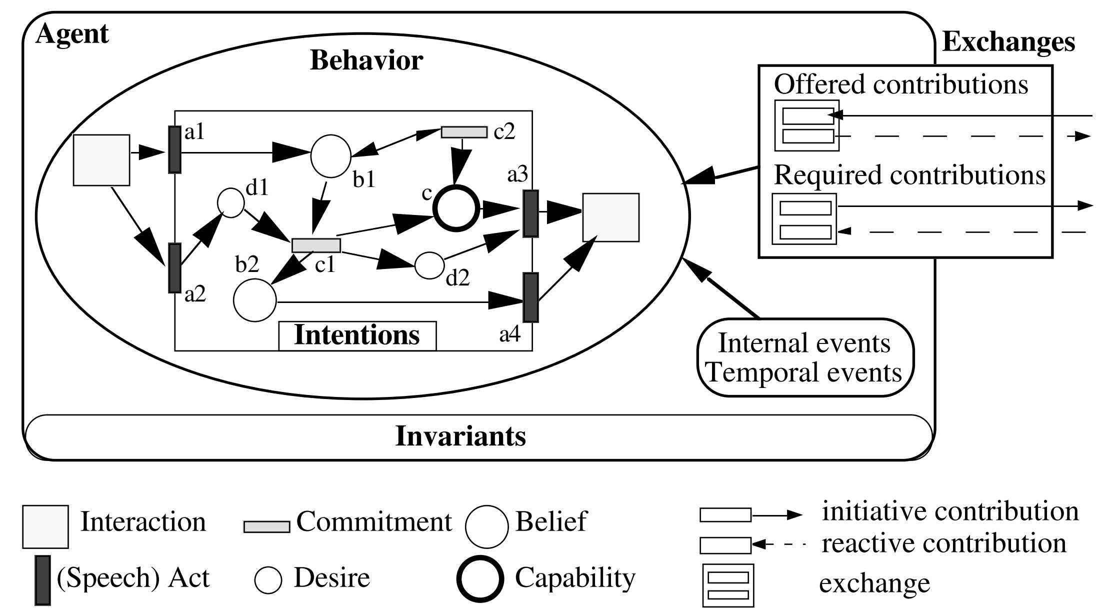


Figure 2 - Agent abstract machine

3.2.1. Intentions

Agent's intentions are expressed by Petri nets (the box "intentions" in figure 4). This Petri net is written with the following rules:

- External transitions (a1,a2,a3,a4) are (speech) acts and permit to act on and from the agent environment depending on intentions' processing;
- Tokens evolving in the Petri net have the format `<f,id,b,t>`. Such token expresses a fact `f` about an agent identified by its identity `id`, is considered to be `b` (true or false) at time `t`;
- Belief is a place (b1,b2) and expresses a condition on commitment or act;
- Desire expresses non satisfied commitments waiting for available belief or capability and are at the input place of commitment or act.
- Commitment (c1,c2) expresses a set of agent internal operations transforming belief and desire into belief, desire or capability. Commitment transition can be hierarchically decomposed into an another time Petri net.

- Capability expresses resource or precondition do act, and are always at the input of an act (c to a3).
- An agent can act from its belief or desire (b2 to a4 or d2 to a3)

3.2.2. Agents' interactions

Agents dialog is viewed as a "*Structural organization of a conversation*" [Sinclair 87]. A conversation is a "text" collectively produced of interlacing elements. Five hierarchical status (figure 3) can be described:

- **Interaction**: it is the higher status rank. It defines the context of a conversation.
- **Transaction**: it is defined as a block of exchanges determinated by the context of interaction.
- **Exchange**: it is the smaller entity of dialogue defining a transaction. It is a bi-directional exchange between two speakers
- **Contribution**: this is an entity sent by an agent.
- **Speech act**: it is the minimal unit of conversational grammar. A speech act combination forms a contribution.

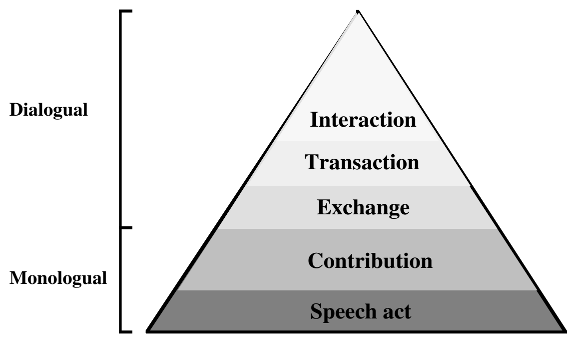
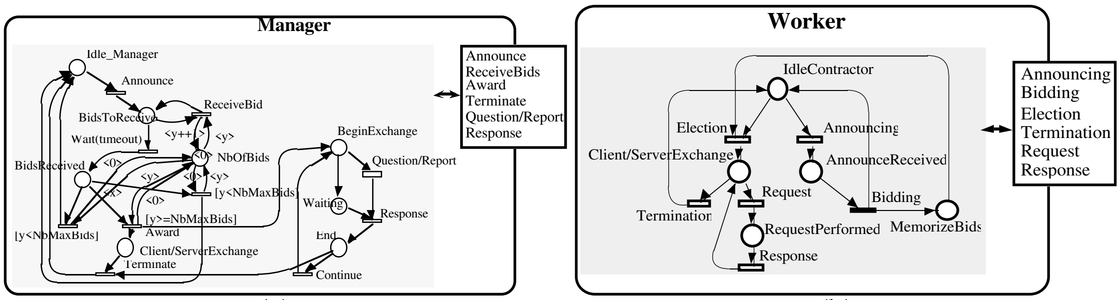


Figure 3 - hierarchical dialogued speech

Task-sharing

The most known related protocol to task-sharing is the Contract-Net [Smith 80]. The figure 4 gives a Petri net model of Manager (figure 6a) and Worker (figure 6b) mechanisms. The contributions of the manager: Announce (Bids), ReceiveBids (Inform), Award (Request), Terminate (Inform), Question/Report (Request), Response (Inform), are expressible in terms of speech acts. The set of possible exchanges of manager are (Announce, ReceiveBids), (Award, Terminate), (Question/Report, Response). The Worker uses the contributions: Announcing (Request), Election (Request) and Bidding (bid). The set of possible exchanges of one worker are (Announcing, Bidding), (Election, Termination) and (Request, Response).



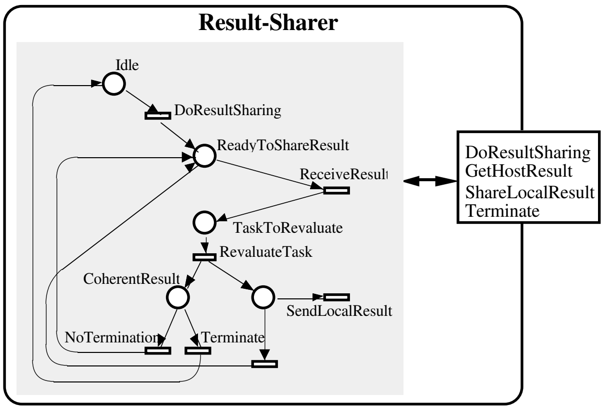
(a) (b)

Figure 4 - task-sharing modeling

Result-sharing

Cooperation by result-sharing [Lesser 81] assumes tasks to be already allocated to agents. Agents commit in an iterative exchange of partial and temporary solutions in order to identify relation about their problems and to converge to consistent local solutions.

The figures 5 gives a Petri net model of Result-sharing mechanisms. The contributions DoResultSharing (Request), GetHostResult (Inform), ShareLocalResult (Inform), Terminate (Request) are expressed in terms of speech acts. The set of possible exchanges are (DoResultSharing, Terminate) and (GetHostResult, ShareLocalResult).



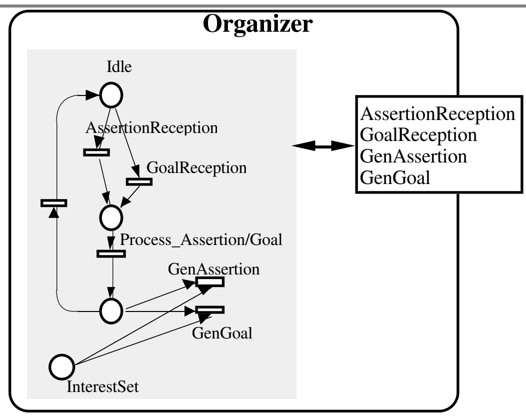


Figure 5 - result-sharing

Figure 6 - organizing

Organizing

Organizing can be seen as a process based on a set of assertions and refutations of questions applied to an intuitive structured organisation (see [Kornfeld 81] for the scientific community metaphor).

The figure 6 gives a Petri net model of organizing mechanisms. The contributions AssertionReception (Inform), GoalReception (Request), GenAssertion (Inform), GenGoal (Request) are speech acts. The set of possible exchanges are

(AssertionReception, GenAssertion), (AssertionReception, GenGoal), (GoalReception, GenAssertion) and (GoalReception, GenGoal).

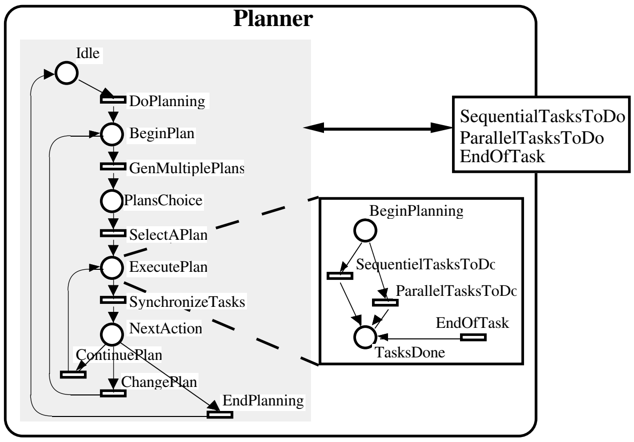


Figure 7 - Planning

Planning

Planning is viewed as an arranged set of operations working out wanted goals, alleging the only changes of the state of the world which are caused by planned operations. The figure 7 gives a Petri net model of centralized Planning mechanisms: the contributions SequentialTasksToDo, ParallelTasksToDo and EndOfTask are speech acts of type Request. The set of possible exchanges are (SequentialTasksToDO, TasksDone) and (ParallelTasksToDo, TasksDone).

3.3. Proofs and Petri nets

To prove precited properties, Petri nets community addresses techniques of formal analysis which consist of studying the structural and behavioral properties of the system without doing simulations or tests.

The occurrence graph [Huber 86, Chiola 90, Valmari 92, Notomi 94], place and transition flows [Couvreur 90], reduction [Haddad 89], performance analysis is done through analysis of time in the Petri net model (stochastic Petri nets) [Dutheillet 90]

4. CONCLUSION

The IAM method suggests requirements capture with intentional agent conceptualization, through interactions and speech acts modeling. The method proposes to refine the analyzed model into a model composed of Petri nets (expressing system dynamic) and temporal logic (expressing system properties to prove). The paper emphasizes the need for property's analysis (such as safety, liveness, performance), proof obligation notions of the system, since the requirement elicitation level. Such properties are proved on the Petri net fmodel using numerous associated verification techniques.

Current tool supports are Petri nets graphic editor, type checking, proof, properties prover, simulator, prototyping. Planning tools are semantic parser, type checking, proof obligation step, interactive prover, agent library manager, incremental specification and design, viewpoints managing (useful in a groupware issue).

This work is in the context of the MARSALA (Modelling, Analysis and Realization of Systems of Agents with Languages Actors based) project [Coriat 94, IBP 92]. This project defines methods for specification and design to build applications using agent conceptualization model, analyzed with Petri nets tools and implemented with the actor calculus model [Agha 86]. The MARSALA field of applications is varying: robotic, AI problems resolution (planning, behavior simulation) [Coriat 93], and groupware [Bachatène 93].

5. REFERENCES

[Abrial 93] J.-R. Abrial, The B method. CNAM courses, 1993.

[Agha 86] G. Agha .- Actors: a Model of Concurrent Computation in Distributed Systems .- Series in Artificial Intelligence, MIT Press, 1986

[Austin 62] J.L. Austin .- How to do things with words" .- Oxford University Press, 1962

[Bachatène 93] H. Bachatène, A. El Fallah Seghrouchni, M. Coriat .- Using Software Engineering Principles to Formalize Cooperative Intelligent Systems .- To appear in proceedings of 5th internationale conference "Software Engeneering and Knowledge Engeneering " , IEEE press, San Fransisco, USA, June 1993.

[Bond 88] Alan H. Bond .- Les Gasser Readings in Distributed Artificial Intelligence .- Morgan Kaufmann Publishers, Inc. San Mateo, California, 1988

[Cohen 90] P.R. Cohen, H.J. Levesque .- Intention Is Choice with Commitment .- A.I. 42, pp 213-261, 1990

[Coriat 93] Coriat M., Stepwise Development of Correct Agents: A behavioural approach based on Coloured Petri Nets. Canadian Conference on Electrical and Computer Engineering 1993 (IEEE eds.)

[Coriat 94] Coriat M., Modeling of Coordinated Agents in a Multi-Agents System. To appear in the proceedings of the 9th Int. Symp. on Computer. and Information Sciences, ISCIS'94, November 7-9, Antalya, Turkey.

[Couvreur 90] Couvreur J.M., Martinez J. Linear invariants in commutative high level nets. In G. Rozenberg (ed.): Advances in Petri nets 1990, LNCS 483, 1991, pp 146-165.

[Dennett 87] D.C. Dennett, The Intentional Stance. The MIT Press, 1987

[Dutheillet 90] C. Dutheillet and S. Haddad .- Regular Stochastic Petri Nets .- G. Rozenberg (eds) : Advances in Petri nets 1990. LNCS, vol. 483. Springer, Berlin Heidelberg New York 1990, pp. 186-210

[Haddad 89] S. Haddad .- A Reduction Theory for Coloured Nets .- G. Rozenberg (eds) : Advances in Petri nets 1989. LLNCS, vol. 424. Springer, Berlin Heidelberg New York 1990, pp. 209-235

[Holm 94] Holm P., The COMMODIOUS Method - Communication Modelling as an Aid to Illustrate the Organisational Use of Software, 6th Int. conf. on Software Engineering and Knowledge Engineering, Jurmalal, Latvia.

[Huber 86] Huber P., Jensen A.M., Jepsen L.O., Jensen K. Reachability trees for high-level Petri nets. Theoritical computer science 45 (1986), North-Holland, 261-292. Also in [Jensen 91] 319-350.

[IBP 92] Estraillier P., Ferber J., B. Habert, F.X. Testard-Vaillant, Rapport d'activités et travaux MASI dans le cadre du projet MARSALA .- IBP report .- December 1992.

[Jensen 91] K. Jensen & G. Rozenberg (eds.),High-level Petri nets, Theory and Applications, Springer Verlag, 91.

[Jensen 92] Jensen K., Coloured Petri nets, basic concepts, analysis methods and practical use, vol 1. Lecture note in computer science, 1992.

[Kornfeld 81] W. A. Kornfeld, C. E. Hewitt .- The Scientific Community Metaphor .- IEEE Transactions on Systems, Man and Cybernetics, MC-11(1):24-33, January 1981

[Lesser 81] V. R. Lesser, D. D. Corkill .- Functionally Accurate, Cooperative Distributed Systems .- IEEE Transactions on Systems, Man and Cybernetics, SMC-11(1):81-96, January 1981

[Murata 89] Murata T., Petri Nets: Properties, Analysis and Applications. Proc. IEEE, vol 77, N° 4, april 1989

[Nellborn 94] Nellborn C., Holm P., Capturing Information Systems Requirements Through Enterprise and Speech Act Modelling. 6th Int. Conf. CAiSE'94, Netherlands, June 94, LNCS 811.

[Notomi 94] M. Notomi, T. Murata, Hierarchical Reachability Graph of Bounded Petri Nets for Concurrent Software Analysis, in IEEE Trans. on Software Engineering, Vol 20, N° 5, May 1994.

[Pnueli 92] Z. Manna, A. Pnueli, The Temporal Logic of Reactive and Concurrent Systems, Springer-Verlag, 1992.

[Shoam 93] Y. Shoam .- Agent-Oriented Programming, Artificial Intelligence Magazine 60, 51-92, March 1993. Also Tech. report STAN-CS-1335-90, Stanford 90

[Sinclair 87] A. Sinclair and R.M. Coulthard .- Towards an Analysis of Discourse. The English used by Teachers and Pupils .- Oxford : Oxford University Press

[Smith 80] R. Smith .- The Contract Net Protocol : High-Level Communication and Control in a Distributed Problem Solver IEEE Transactions on Computers C-29(12):1104- 1113, december 1980

[Valmari 92] Valmari A. Stubborn sets of coloured Petri nets. In G. Rozenberg (ed.): Advances in Petri nets 1992, LNCS 1992.

[Wirfs-Brock 90]R. Wirfs-Brocks, B. Wilkerson, L. Wiener. Designing Object-Oriented Software. Prentice-Hall, 1990.

[Wooldridge 94] M. Wooldridge, The logical Modelling of Computationnal Multi-Agent Systems. PhD thesis, January 1994. Tech report MMU-DOC-94-01

[Yu 94] Eric S. K. Yu and John Mylopoulos. Understanding ``Why'' in Software Process, Modelling, Analysis, and Design (Research Paper). Proceedings of 16th International Conference on Software Engineering, May 16-21, 1994, Sorrento, Italy, pp. 159-168.