The multi-agent programming paradigm use for railway applications

F. M. Rachel & P. S. Cugnasca
Safety Analysis Group, Department of Computer Engineering and Digital Systems, Polytechnic School, University of São Paulo, Brazil

Abstract

As railway signalling systems hardware improved from mechanical switches and electrical relays to electronic tubes and semiconductors, the software improved from direct control and event-driven approach to object and agent-oriented programming.

This paper aims at the use of agent-oriented programming for railway applications. In order to do that some agent basic concepts such as organizations (reactive and cognitive), models (beliefs/desires/intentions and social) and interactions (independent and collaborative) between agents are presented.

The AOSE (Agent Oriented System Engineering), as OOSE (Object Oriented System Engineering) some time ago, is a paradigm on multi-agent programming. The main difference between an agent and an object is that the agent must have autonomy while an object must not. An object property is only activated when a control program asks for that property. An agent, instead, must start and manage its own properties in order to perform its objectives.

Therefore, a model proposal for train control systems is presented based on multi-agent programming paradigm. In this proposal, trains and stations are modelled as interacting agents, working together in order to achieve a common goal: an optimized train traffic control. Some profits and drawbacks of this approach are also addressed, detaching the importance of multi-agent approach characteristics for railway and train control applications. The AOSE supplies many tools for multi-agent systems design and programming.

Keywords: railway signalling systems, train control, multi-agent systems, multi-agent programming, multi-agent programming paradigm, AOSE (Agent Oriented System Engineering).

WIT Transactions on The Built Environment, Vol 103, © 2008 WIT Press
www.witpress.com, ISSN 1743-3509 (on-line)
doi:10.2495/CR080621

1 Railway signalling systems

The railway signalling systems hardware and software, as other control systems, have improved along time according to technology evolution.

In this way, the first railway signalling was made up of mechanical signals, parts and switches. Later, some parts were replaced by electrical relays as such switching elements. With electronic evolution, tubes were initially introduced in railway signalling systems hardware and were replaced by semiconductors, parts with improved reliability and less energy consumption [1].

As semiconductors integration scale has become increasingly high, the IC (Integrated Circuits) were introduced in control systems hardware, evolving from logical elements to microprocessors, micro-controllers and PLC (Programmable Logic Controllers) [1].

On the other hand, the railway signalling software also evolved from wired connections, on the first electrical relays systems to the first direct control software, composed by an operational command sequence for relay control. With the introduction of microprocessors, micro-controllers and PLC in railway signalling systems hardware, the first programming languages such as assembly were introduced [2].

Later, some higher-lever languages, as such C language were introduced in system design and development to make the system programmers´ and developers´ work easier. The software was initially recorded in punched paper tape and tape cards and later these records were replaced by magnetic media [2].

The system development process has also evolved from a central structured analysis to an Object-Oriented approach. This evolution was named Object-Oriented paradigm. In this paradigm, the tasks were distributed along the many objects designed to compose the system. The control program had to activate the objects, selecting their desired properties. Other objects were used to sense some operational conditions, communicating them to the control program, which would take some pre-defined actions [2].

A proposal for an object-oriented automatic train controller was introduced and presented in Comprail 2004 and Comprail 2006 in papers called “Object-oriented Approach for Automatic Train Operation Control Systems” and “Using UML Diagrams for System Safety and Security Environment Analysis”, respectively [2].

This paper introduces and proposes a new program paradigm to be used in railway applications: the multi-agent systems programming. In this kind of system, agents are used instead of objects. Figure 1 shows the railway signalling systems hardware and software evolution along time [2].

In the following sections, the multi-agent systems are introduced as well as the multi-agent programming paradigm. Then, the idea for an automatic train controller using multi-agent systems is introduced and a study case for trains updating process in the São Paulo Subway Company is presented, leading to some conclusions about the multi-agent program paradigm use in railway and other control systems applications.

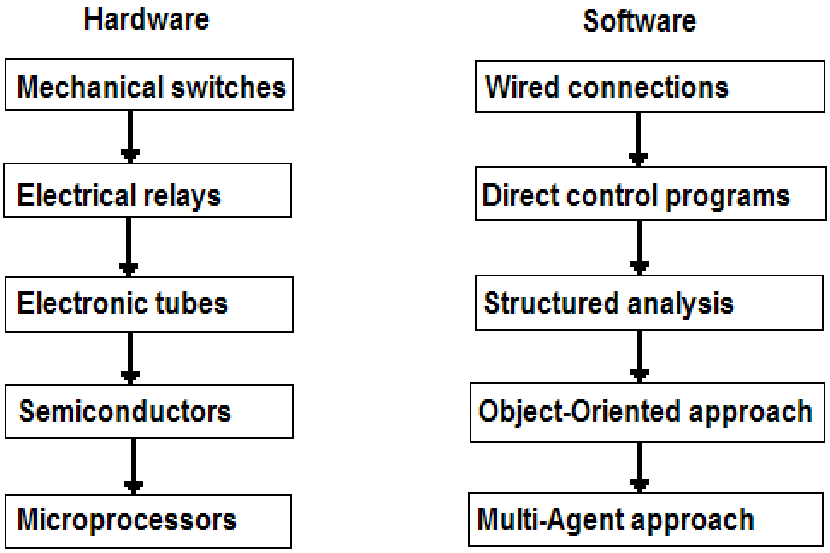

Figure 1: Railway signalling hardware and software evolution.

2 Multi-agent systems

An agent is defined as any hardware or software element designed specifically to perform pre-determined tasks. An agent must have the following properties: autonomy, sociability, reactivity and initiative.

A multi-agent system is defined as a system composed by agents organized as a structure inside an environment, interacting with each other to perform a determined objective. An example of a multi-agent system is a soccer team, where each soccer player has a pre-determined task or role to be performed (such as goalkeeper, back, middle and attackers); however, the team as a whole has an objective (win the game) and the players interact with each other to score a goal and win the game [3].

According to the structure characteristics, the multi-agent systems can be classified as reactive or cognitive. In the reactive multi-agent systems, the agents do not interact with each other in a direct way, but they interact with the environment, and the change in the environment is sensed by the other agents. In the cognitive multi-agent systems, the agents communicate with each other in a direct way, so that they know the presence and the characteristics of all the agents that compose the systems. Table 1 shows the structure characteristics for multi-agent systems [4].

The reactive multi-agent systems are usually ethologic systems composed by many agents, which have the same characteristics. In this kind of system, there is not an implicit environment representation neither tasks registration nor tasks deliberation [4].

Table 1: Multi-agent systems structure characteristics.

Characteristic	Reactive	Cognitive
Environment representation	No	Implicit
Task Registration	No	Yes
Task deliberation	No	Yes
Type of system	Ethologic	Social
Quantity	Many	Few
Agent characteristics	Equal	Different
Communication	Direct	Indirect

The cognitive multi-agent systems are usually social organization of few agents, which have different characteristics. In this kind of system, there is an implicit environment representation, tasks registration and tasks deliberation. However, the reactive multi-agent systems implementation is much easier than cognitive ones due to the fact that in reactive multi-agent systems there is no need of a communication channel implementation between the agents [4].

As the agents are grouped together, they make a social structure. The next step is to program each agent with an expected behaviour that makes use of the social reasoning over the individual reasoning. In this social reasoning, the interaction aspects must be considered, as agent communication, interaction and negotiation. The main aspect of multi-agent systems use is to take advantage of the collaborative behaviour that emerges from agent interaction [4].

According to the social interaction characteristics, the multi-agent systems can be classified as collaborative or independent. In collaborative systems, the agents have flexible tasks that can be changed to perform a social objective. In independent systems, the agents have fixed tasks that must be coordinated with other agents´ tasks to perform the social objective. Table 2 shows the structure characteristics for multi-agent systems [4].

Table 2: Multi-agent systems social interaction characteristics.

Characteristic	Independent	Collaborative
Type of tasks	Fixed	Flexible
Communication	No	Needed
Control	Coordination	Interaction
Global behaviour	Sum	Interaction

To help the agent programming tasks, there are some theories to model the agent behaviour as function of its beliefs, desires and intentions, called BDI theories. In these theories, the agent behaviour is modelled according to its environment view (beliefs), its objectives (desires) and its reactions (intentions) [4].

3 Multi-agent programming paradigm

The AOSE (Agent Oriented System Engineering) emerged from the need of a methodology for multi-agent systems implementation and includes modelling and programming tools for multi-agent systems.

The multi-agent systems can be modelled in two ways: the agent-centred and the organization-centred point of view. In the agent-centred point of view, the social concepts are focused on the agent behaviour, as a social entity. Some tools that support this point of view are: MANTA (Modelling an ANT hill Activity), in which the model is based on the work division within a primitive ant society, CNET (Contract NET), where each agent seeks help from other agents to perform its tasks and DBC (Dependence Based Coalitions), where the social needs are communicated to and from all agents by means of announcements and bids and coalitions between agents are established [5].

In the organization-centred point of view, the social concepts are focused on the organization objectives, which constrains the agent behaviour. Some tools that support this point of view are: GAIA, in which an organization goal is translated into requirements specifications that specify the agent organization and behaviour, TAEMS (Task Analysis, Environment, Modelling and Simulation), in which top-level goals are established to coordinate agent tasks, AGR (Agent Group Role), in which the agent group and structure are defined based on roles to be performed by agents, STEAM (Shell for TEAMwork), in which a general framework is defined to enable agents to participate in teamwork and MOISE (Model of Organization for multI-agent SystEms), in which three main dimensions of the organization are specified: structural, functional and relational [5].

The AOSE also supports some tools for agent-oriented programming languages. Some of these languages are: JASON, a Java interpreter and a logic-based agent-oriented programming language as an extended version of AgentSpeak, JADE (Java Agent Development Environment), an open source project distributed by TILab (Telecom Italia Labs) and JACK, an agent development environment produced by the Agent Oriented Software Group [5].

Now, an example of a multi-agent system modelling will be presented. Using the MOISE, a soccer team will be modelled. The MOISE methodology establishes that three main dimensions of the organization must be specified: the structure, the functional end the relational dimension. In structure dimension, the agent desired behaviour is translated into roles to be performed by the agents. In this way, the different roles and their interrelations are specified. Figure 2 shows the structure of a soccer team [6].

In functional dimension, the organizational goals are specified and divided into sub-goals. In this specification, the missions and the sequence of the tasks are also detailed. Figure 3 shows the social scheme for the objective “score a goal”. Table 3 shows the goals descriptions for Figure 3 [6].

In relational (also called deontic) dimension, the relations between agents are specified in individual level as permissions and obligations of a role on a mission [6].

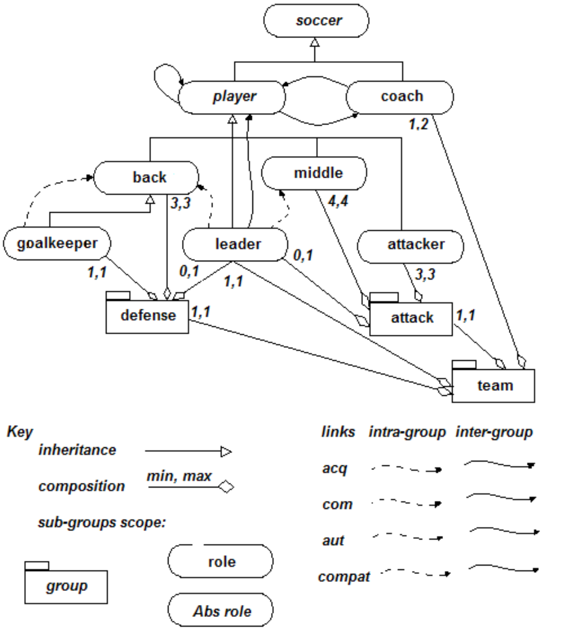

Figure 2: Structure of a soccer team [6].

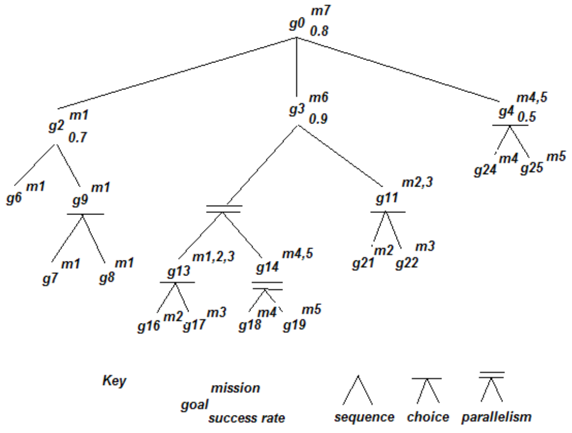

Figure 3: Social scheme to the objective “score a goal” [6].

4 Train control systems

Train control systems are intended to control train traffic along the commercial line. According to the signalling system, train control can be classified as fixed and mobile blocks. In fixed blocks signalling systems, the main track is divided

Table 3: Goal description of Figure 3 [6].

goal	Description
g_0	Score a soccer goal
g_2	The ball is in the middle field
g_3	The ball is in the attack field
g_4	The ball was kicked to the opponent's goal
g_6	A team mate has the ball in the defence field
g_7	The ball was passed to a left middle
g_8	The ball was passed to a right middle
g_9	The ball was passed to a middle
g_{11}	A middle passed the ball to an attacker
g_{13}	A middle has the ball
g_{14}	The attacker is in good position
g_{16}	A left middle has the ball
g_{17}	A right middle has the ball
g_{18}	A left attacker is in a good position
g_{19}	A right attacker is in a good position
g_{21}	A left middle passed the ball to a left attacker
g_{22}	A right middle passed the ball to a right attacker
g_{24}	A left attacker kicked the ball to the opponent's goal
g_{25}	A right attacker kicked the ball to the opponent's goal

into track circuits, that is, segments of tracks on which it is possible to locate a train and transmit to it information such as a maximum speed code allowed in that track. Figure 4 shows a fixed block signalling system [7].

In mobile blocks signalling systems, there is no fixed length track circuits and the distance between trains is controlled using the information of relative distance and speed between trains. Using this information, the train aboard equipment controls train propulsion or braking, according to the distance and the next train speed [7].

There are two ways of implementing a mobile block signalling: the centralized control, in which a central equipment receives each train position and speed and transmits to each train the maximum speed allowed, according to its distance and relative speed to the next train and the distributed control, in which each train itself, using the information of relative distance and speed transmitted by the next train, controls its propulsion or braking, according to the information received [7].

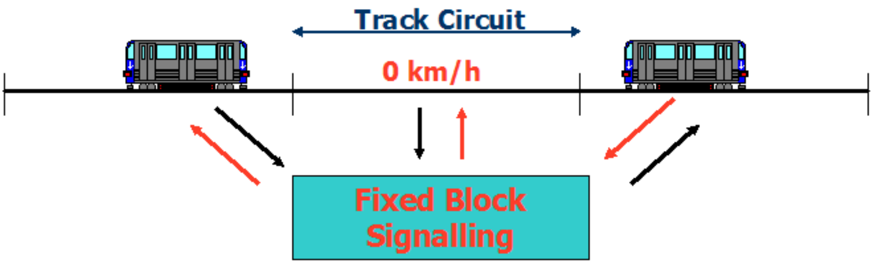

Figure 4: Fixed block signalling system.

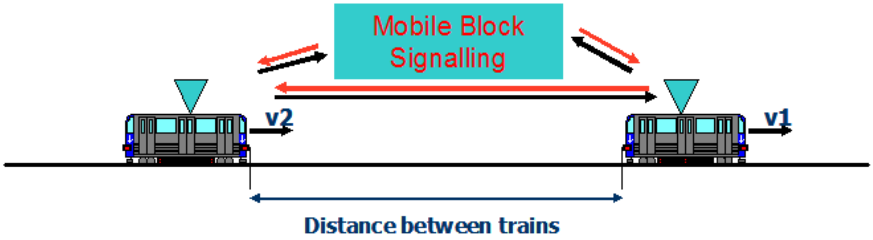

Figure 5: Centralized mobile block signalling system.

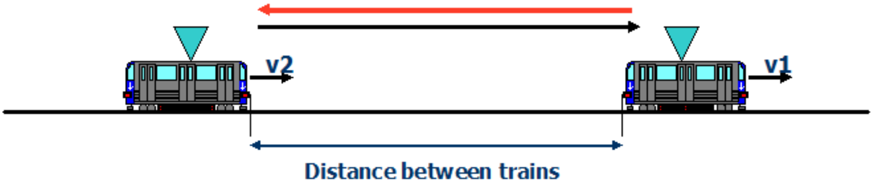

Figure 6: Distributed mobile block signalling system.

In both cases, there is the need of a communication channel implementation between trains or between the trains and the centralized controller. The idea is to take advantage of this communication channel implementation to set a cognitive and collaborative multi-agent system. In this proposed system, each train and each station along the line would be modelled as an agent, so the stations passenger demand could be transmitted to the trains in order to optimize train supply according to the passenger demand needed. Thus, the operational train control would be much easier than the control used now.

4.1 Study case: São Paulo Subway Company

In the São Paulo Subway Company there is a project for trains updating, since the first operational fleet is about thirty-five years old. This project includes the changing of the signalling system from fixed to mobile blocks using CBTC (Communication Based Train Control).

At the moment, the system specifications have been developed in order to hire a company for design and implementation of the CBTC systems, besides the braking and door opening and closing systems updating, including air conditioner and some mechanical adaptation to the cars structures to optimize the passengers′ space.

The idea is to include in this specification the multi-agent system implementation. As it does not require hardware for its implementation, the costs will only be in the software development. The main advantage of multi-agent systems is that they will not cause any impact to the system implementation, since it can be made with CBTC implementation.

5 Conclusions

The multi-agent system is a distributed AI (Artificial Intelligence) tool suitable for situations in which the solution for a problem is too complex to use direct control tools. The multi-agent systems allow the division of complex problems into smaller ones and the solution emerges from the cooperative behaviour between the system compound agents.

The fact that to implement a cognitive and collaborative multi-agent system it is necessary to implement a communication channel between the agents can be added to the fact that mobile signalling systems also make use of a communication channel between trains, making it far easier to implement a multi-agent system together with a mobile signalling system, as such CBTC. In the case of the São Paulo Subway Company, the implementation of multi-agent system together with trains updating process, particularly together with CBTC implementation, is a low-cost implementation, because the multi-agent system does not require special hardware and the costs are related to multi-agent software development.

This AI tool is suitable for use in railway applications, because it can be used for controlling the railway traffic in an efficient way. However, so as not to disregard European standards, this tool can only be used in automatic train operation (ATO) functions. The ATP functions must be implemented without the use of AI tools, and the communication channel used for transmitting and receiving trains positions and speeds must be apart of that used for agent communication. There are many safety aspects that must be considered on ATP functions implementation, such as the communication channel redundancy, the increase in train position measuring accuracy and the importance of knowing the real braking profile, among others.

References

[1] United States Congress – Office of Technology Assessment, *Automatic Train Control in Rail Rapid Transit (appendix E), Chronology of Train Control Development.* http://www.ww.princeton.edu/cgi-bin/byteserv.prl /~ota/disk3/1976/7614/761414.pdf

[2] Rachel, F.M., Cugnasca, P.S., *Object-oriented approach for automatic train operation control systems,* Computers on Railways IX Proceedings, WitPress, 2004.

[3] Wooldridge, M., Jennings, N. *Introduction to Multi-Agent Systems,* ICMAS 95 Workshop on Introduction to Multi-Agent Systems, 1995.

[4] Demazeau, *Distributed AI and Multi-Agent Systems*, Lecture notes, University of Odense, Denmark, 1994.

[5] Wooldridge, M, Weiss, G. et al. *Agent-Oriented Software Engineering*, Springer-Verlag, Berlin.

[6] Hubner, J.F., Sichman, J.S. and Boissier, O. *A Model for the Structural, Functional and Deontic Specification of Organizations in Multiagent Systems*, 2002.

[7] Rachel, F.M, *Proposta de um Controlador Automático de Trens Utilizando Lógica Nebulosa Preditiva*, Universidade de São Paulo, São Paulo, 2006.