Organizational Multi-Agent Architectures: A Mobile Robot Example

Manuel Kolp
IAG School of Management
University of Louvain
1, Place des Doyens
B-1348, Louvain-La-Neuve - Belgium
Tel.: +32 10 47 83 95
kolp@isys.ucl.ac.be

Paolo Giorgini
Department of Mathematics
University of Trento
Via Sommarive, 4
I-38050, Trento – Italy
Tel.: +39 0461 882052
pgiorgini@science.unitn.it

John Mylopoulos
Department of Computer Science
University of Toronto
6 King's College Road
M5S 3H5, Toronto - Canada
Tel.: +1 416 978 5180
jm@cs.toronto.edu

ABSTRACT
We propose architectural styles for multi-agent systems motivated by organization theory. One of them is discussed in the paper.

Categories and Subject Descriptors
D.2.1 [**Software Engineering**]: Requirements/Specifications – *elicitation methods, languages, methodologies*; D.2.11 [**Software Engineering**]: Software Architectures – *data abstraction, patterns*; K.6.1 [**Management of Computing and Information systems**]: Project and People Management – *systems analysis and design*; K.6.3 [**Management of Computing and Information systems**]: Software Management – *software development*.

General Terms: Design

1. MOTIVATION
Software architectures describe a software system at a macroscopic level in terms of a manageable number of subsystems/components/modules inter-related through data and control dependencies. Software architectures have been the focus of considerable research for the past decade which has resulted in a collection of well-understood architectural styles. Examples of styles are pipes-and-filters, event-based, layered and the like [3].

We are interested in developing a suitable set of architectural styles for multi-agent software systems. Since the fundamental concepts of multi-agent systems are intentional and social, rather than implementation-oriented, we turn to theories which study social structures for motivation and insights. But, what kind of social theory should we turn to? There are theories that study group psychology, communities and social networks. Such theories study social structure as an *emergent property* of a social context. Instead, we are interested in social structures that emerge from a design process. For this, we turn to organizational theory for guidance.

AAMAS'02, July 15-19, 2002, Bologna, Italy.

2. ORGANIZATIONAL STYLES
Organizational theory studies alternative styles for (business) organizations. These styles are used to model the coordination of business stakeholders -- individuals, physical or social systems -- to achieve common goals. Each organizational style represents a possible way to structure an organization in order to meet its strategic objectives.

The structure of an organization defines the roles of various intentional components (actors), their responsibilities, defined in terms of tasks and goals they have been assigned, and resources they have been allocated. Moreover, an organizational structure defines how to coordinate the activities of various actors and how they depend on each other. Such dependencies may involve both actors of the organization and its environment (e.g., partners, competitors, clients, etc.).

An organizational style defines a class of organizational structures, and offers a set of design parameters that can influence the division of labor and the coordination mechanisms, thereby affecting how the organization functions. Design parameters include, among others, task assignments, standardization, supervision and control. The organization designer can use these parameters in order to deal with both *situational* and *contingency factors*, namely organizational states or conditions that are associated with the use of certain design parameters. Contingency factors may involve age and size of the organization, its technical infrastructure, as well as characteristics of the environment, such as stability, complexity, diversity, and hostility.

We propose a macro level catalogue of styles adopting (some of) the abstractions offered by organization theory for designing multi-agent architectures. For further detail see [1]. We describe here the structure-in-5 and model it using *i**.

An *i** model [4] is a graph, where each node represents an actor (an agent, position, or role) and each link between two actors represents a social dependency. Such a dependency can represent the fact that one actor depends on another for a goal to be fulfilled, a task to be performed, or a resource to be made available. The depending actor is called the *depender* and the actor who is depended upon the *dependee*. The object around which the dependency centers (goal, task or resource) is called the *dependum*. The model distinguishes between goals, which are well defined, and softgoals, which do not have a formal definition and are amenable to a different (more qualitative) kind of analysis.

For instance, in Figure 1, the *Technostructure, Middle Agency* and *Support* actors depend on the *Apex* for strategic management. Since the goal *Strategic Management* does not have a precise description, it is represented as a softgoal (cloudy shape). The *Middle Agency* depends on the *Technostructure* and *Support* respectively through goal dependencies *Control* and *Logistics* represented as oval-shaped icons. The *Operational Core* is related to the *Technostructure* and *Support* actors through the *Standardize* task dependency and the *Non-operational Service* resource dependency, respectively.

The structure-in-5 (Figure 1) is a typical organizational style. At the base level, the *Operational Core* takes care of the basic tasks — the input, processing, output and direct support procedures — associated with running the organization. At the top lies the *Apex*, composed of strategic executive actors. Below it, sit the *Coordination*, *Middle Agency* and *Support* actors, who are in charge of control/standardization, management and logistics procedures, respectively. The *Coordination* component carries out the tasks of standardizing the behavior of other components, in addition to applying analytical procedures to help the organization adapt to its environment. Actors joining the apex to the operational core make up the *Middle Agency*. The *Support* component assists the operational core for non-operational services that are outside the basic flow of operational tasks and procedures.

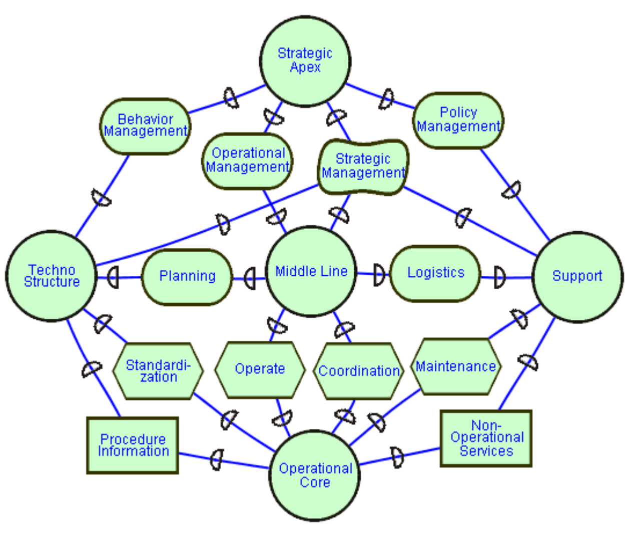

Figure 1. Structure-in-5

3. A MOBILE ROBOT ARCHITECTURE

Consider the following activities [3] an office delivery mobile robot typically has to accomplish: acquiring the input provided by sensors, controlling the motion of its wheels and other moveable part, planning its future path. In addition, a number of factors complicate the tasks: obstacles may block the robot's path, sensor inputs may be imperfect, the robot may run out of power, mechanical limitations may restrict the accuracy with which the robot moves, the robot may manipulate hazardous materials, unpredictable events may leave little time for responding.

Figure 2 depicts a structure-in-5 robot architecture in *i**. The *moveable parts controller* component is the *operational core* managing the robot motors, joints, wheels, etc. The *Global Planner* is the *strategic apex* planning and scheduling the robot's mission. The *sensors* compose the *support* component *capturing real world raw information* from hardware multiple sensors and integrating it into a coherent real-time interpretation for the *Navigator* component. It also gives direct *external* feedback to the *Moveable Parts Controller*. The *Real World Modeler* is the *technostructure* concerned with *planning* the mission *paths*, establishing and maintaining the *robot's model* of the *world* and checking the robot's mission environment to ensure *predictability management*. The *Navigator* is the *middle agency* component, the central intermediate module *coordinating the movements* of the robot to assume *failability tolerance* and *adaptability management*.

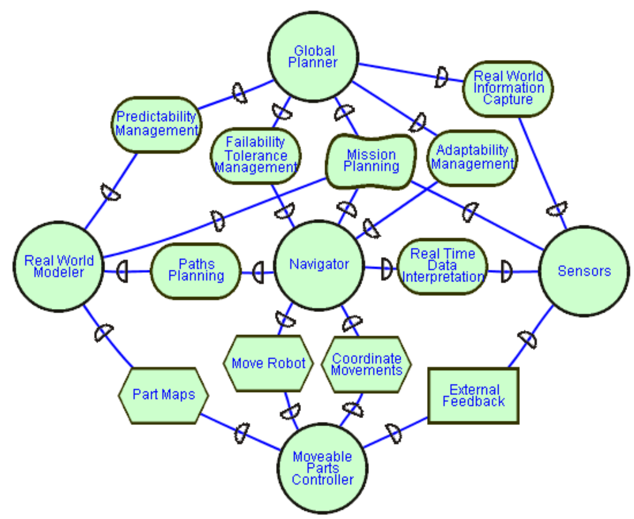

Figure 2. A structure-in-5 mobile robot architecture.

4. REFERENCES

[1] Kolp, M., Giorgini P., and Mylopoulos J. "An Organizational Perspective on Multi-agent Architectures". In *Proc. of the Eighth Int. Workshop on Agent Theories, architectures, and languages*, *ATAL'01*, Seattle, USA, August 1-3, 2001.

[2] Scott, W. R. Organizations: Rational, Natural, and Open Systems, Prentice Hall, 1998.

[3] Shaw, M., and Garlan, D. *Software Architecture: Perspectives on an Emerging Discipline*, Upper Saddle River, N.J., Prentice Hall, 1996.

[4] Yu, E. *Modelling Strategic Relationships for Process Reengineering*, Ph.D. thesis, Department of Computer Science, University of Toronto, Canada, 1995.