

Chapter 14

Design Patterns for Social Intelligent Agent Architectures Implementation

Manuel Kolp

Université catholique de Louvain, Belgium

Yves Wautelet

KU Leuven, Belgium

Samedi Heng

Université catholique de Louvain, Belgium

ABSTRACT

Multi-agent systems (MAS) architectures are popular for building open, distributed, and evolving software required by today's business IT applications such as e-business systems, web services, or enterprise knowledge bases. Since the fundamental concepts of MAS are social and intentional rather than object, functional, or implementation-oriented, the design of MAS architectures can be eased by using social patterns. They are detailed agent-oriented design idioms to describe MAS architectures as composed of autonomous agents that interact and coordinate to achieve their intentions like actors in human organizations. This chapter presents social patterns and focuses on a framework aimed to gain insight into these patterns. The framework can be integrated into agent-oriented software engineering methodologies used to build MAS. The authors consider the broker social pattern to illustrate the framework. The mapping from system architectural design (through organizational architectural styles), to system detailed design (through social patterns), is overviewed with a data integration case study.

1. INTRODUCTION

This section introduces and motivates the research. In Section 1.1, we describe the advantages of using multi-agent systems over traditional systems. Section 1.2 presents the importance of *patterns* for designing information systems. We formulate our research proposal in Section 1.3. The Section 1.4 introduces elements for work validation. The context of the research and an overview of the state of the art are given in Section 1.5. Finally, Section 1.6 presents the organization of the chapter.

1.1. Advantages of Multi-Agent Systems

The meteoric rise of Internet and World-Wide Web technologies has created overnight new application areas for enterprise software, including eBusiness, web services, ubiquitous computing, knowledge management and peer-to-peer networks. These areas demand software that is robust, can operate within a wide range of environments, and can evolve over time to cope with changing requirements. Moreover, such software has to be highly customizable to meet the needs of a wide range of users, and sufficiently secure to protect personal data and other assets on behalf of its stakeholders.

Not surprisingly, researchers are looking for new software designs that can cope with such requirements. One promising source of ideas for designing such business software is the area of multi-agent systems. Multi-agent system architectures appear to be more flexible, modular and robust than traditional including object-oriented ones. They tend to be open and dynamic in the sense they exist in a changing organizational and operational environment where new components can be added, modified or removed at any time.

Multi-agent systems are based on the concept of agent which is defined as “a software component situated in some environment that is capable of flexible autonomous action in order to meet its design objective” (Aridor & Lange, 1998). An agent exhibits the following characteristics:

- **Autonomy:** An agent has its own internal thread of execution, typically oriented to the achievement of a specific task, and it decides for itself what actions it should perform at what time.
- **Situatedness:** Agents perform their actions in the context of being situated in a particular environment. This environment may be a computational one (e.g., a Web site) or a physical one (e.g., a manufacturing pipeline). The agent can sense and affect some portion of that environment.
- **Flexibility:** In order to accomplish its design objectives in a dynamic and unpredictable environment, the agent may need to act to ensure that its goals are achieved (by realizing alternative plan). This property is enabled by the fact that the agent is autonomous in its problem solving.

An agent can be useful as a stand-alone entity that delegates particular tasks on behalf of a user (e.g., a learning environment in a Massive Open Online Course (Wautelet et al., 2016a), or a goal-driven office delivery mobile device (Castro, Kolp & Mylopoulos, 2002)). However, in the overwhelming majority of cases, agents exist in an environment that contains other agents. Such environment is a multi-agent system (MAS).

In MAS, the global behavior derives from the interaction among the constituent agents: they cooperate, coordinate or negotiate with one another. A multi-agent system is then conceived as a society of autonomous, collaborative, and goal-driven software components (agents), much like a social organiza-

24 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/design-patterns-for-social-intelligent-agent-architectures-implementation/261032

Related Content

Theory Driven Modeling as the Core of Software Development

Janis Osisand Erika Nazaruka (Asnina) (2021). *Research Anthology on Recent Trends, Tools, and Implications of Computer Programming* (pp. 88-107).

www.irma-international.org/chapter/theory-driven-modeling-as-the-core-of-software-development/261023

Some Key Topics to be Considered in Software Process Improvement

Gonzalo Cuevas, Jose A. Calvo-Manzanoand Iván García (2018). *Computer Systems and Software Engineering: Concepts, Methodologies, Tools, and Applications* (pp. 134-160).

www.irma-international.org/chapter/some-key-topics-to-be-considered-in-software-process-improvement/192875

The Formalization of CAME Architecture

Ajantha Dahanayake (2001). *Computer-Aided Method Engineering: Designing CASE Repositories for the 21st Century* (pp. 59-94).

www.irma-international.org/chapter/formalization-came-architecture/6875

Digital Home: A Case Study Approach to Teaching Software Engineering Concepts

Salamah Salamah, Massood Towhidnejadand Thomas Hilburn (2018). *Computer Systems and Software Engineering: Concepts, Methodologies, Tools, and Applications* (pp. 1284-1299).

www.irma-international.org/chapter/digital-home/192923

Congestion Management Using Hybrid Particle Swarm Optimization Technique

Sujatha Balaramanand N. Kamaraj (2012). *Computer Engineering: Concepts, Methodologies, Tools and Applications* (pp. 710-725).

www.irma-international.org/chapter/congestion-management-using-hybrid-particle/62474