# Improved Template for Agent Pattern Description

Ayodele Oluyomi Shanika Karunasekera Leon Sterling
*Department of Computer Science and Software Engineering*
*The University of Melbourne*
*111 Barry Street, Carlton, Victoria 3053, Australia*
*{aoo, shanika, leon}@cs.mu.oz.au*

## Abstract

*Agent-oriented patterns have been considered potentially useful for improving communication and comprehension of the concepts they describe and for bridging the gap between conceptual modeling and implementation. A pattern needs to be appropriately described for it to be an effective tool of experience sharing. However, many existing approaches to agent patterns lack features that we consider important to pattern descriptions. In this paper, we present a template that we believe improves on current ways of describing agent patterns. We also describe the BDI agent architectural level pattern using our template.*

## 1. Introduction

Patterns are an expression of issues such as what works, lessons learnt and mistakes to avoid. Within the agent-oriented programming community, patterns have been suggested as potentially useful for; improving communication and comprehension of the concepts they describe [4], and bridging the gap between conceptual modeling levels and implementation [7].

Our review of various approaches to agent patterns finds that many of these existing approaches lack features we consider important to pattern descriptions, such as problem, context, etc. Most agent pattern descriptions do not reflect appropriate notions of agency [3]. These limitations hinder the usefulness of agent patterns to encourage adoption of agent technology in industry.

In this paper, we present an improved template for agent patterns at a high level architectural level.

The paper is organized as follows. Section 2 describes related work. Section 3 presents our template design. Section 4 describes a pattern (BDI) using our template. Section 5 compares our template with two other templates. Section 6 concludes the paper.

## 2. Related Work

A good pattern description should include a specific recurring *problem* in a *context* that defines a set of *forces* which are resolved by a general *solution* to create a *resulting context*.

Different pattern templates have been used in describing agent oriented patterns. Tahara et al [2] describe a set of patterns with a three element template. Sauvage [3] uses a pattern template with eight elements. Weiss [4] describes individual patterns with a pattern template of five elements. Aridor and Lange [5] use an abridged (six elements) GoF form. Schelfthout et al [7] use an adapted version (seven elements) of the GoF form. Malyankar [9] proposes a pattern template by adding elements and sub categories to the GoF template.

A review of these templates reveals these inadequacies.

- Most existing templates are lacking in agent concepts.
- Lack of clear rationale for the introduction of template elements or shortening existing templates.
- Templates are not tailored to particular levels of abstraction.
- Ambiguity in the meaning of template elements.

We argue that an agent pattern template should be defined with elements that capture the notions of agency and according to the levels of agent system development. The definition should also be in line with the main features of a good pattern to facilitate communication, adaptation and integration. We use a template for agent architectural patterns in grounding these arguments.

## 3. Approach to Pattern Template Design

Our approach for developing different templates for the different categories of patterns is to generate sub-elements for the main features of a good pattern. Therefore, only the sub-elements vary from one pattern category to another where necessary.

Proceedings of the 2005 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'05)
0-7695-2416-8/05 $20.00 © 2005 IEEE

Sub-elements considered appropriate for a particular category of patterns are defined by carrying out two tasks. First is a study of the different levels of agent oriented design to define the design issues and features of software agents peculiar to each level [10]. Second is the examination of some agent oriented design patterns in order to find out what information is crucial to the description of agent patterns.

The emphasis of our approach is to focus on the Forces and Solution elements. A good pattern is one that describes how a general solution is able to resolve a system of forces. Hence, to improve the pattern template, we introduce agent concepts as sub elements to Forces and Solution. Laying out the forces explicitly helps to describe the unresolved forces in the resulting context.

### 3.1. Template for Agent Architectural Patterns

We present a template for describing patterns at an agent architectural level.

**Name:** Describes a name for referring to the pattern and other names by which the pattern is described, if any.

**Classification:** Describes the position of the pattern in the two way classification according to [10].

**Problem:** Defines the recurring problem that the general solution is defined to solve.

**Context:** Describes a particular setting for the problem which provides the basis for defining the relevant forces.

**Forces:** Describes the constraints that are relevant to a particular problem based on the context of the problem. The following agent concepts are introduced as sub elements to the Forces element.

*Goal* defines commitment to goals; hierarchy of goals; stability of goals, etc.

*Autonomy* defines the degree of dependence on client for final decisions; and degree of control over own services

*Social ability* defines the interaction related constraints. Agent interaction may require negotiation strategies; secured messages e.g. in military applications, and so on.

*Environment* defines the nature of agent's environment.

*Adaptive behaviour* defines the need to be able to sense changes in agent's usage trends and behaviour of the environment and modify behaviour accordingly.

*Intelligence* defines the constraints that relate to the amount of knowledge to be used in decision making.

*Decision and action* defines the level of accuracy and urgency that is required in decision making. Also, it defines constraints related to action utilities and costs.

**Solution:** Describes the general solution that best resolves the forces identified. We introduce the following sub elements to capture the notions of agency in the template:

*Control* coordinates the other components of the architecture. It may stand alone or be part of another component. Control handles issues like choice of reactive over deliberative actions (or vice versa). It also handles the autonomy of the agent.

*Strategy* addresses two issues. One is goal definition strategy i.e. how the agent decides the goals to commit to. The other is action execution strategy i.e. how the agent achieves the goal(s). Action execution strategy addresses the decision on reactive or deliberative behaviour or both.

*Knowledge Management* addresses what constitutes domain knowledge; knowledge updates; and decisions about persistence of knowledge.

*Interaction Management* handles interaction issues like type of interaction; strategy for interaction; message interpretation, interaction protocols and so on.

*Environmental Interface* addresses the perception, action and message communication functions of the agent.

The following sub elements have been used in existing pattern templates, however, we classify them as sub elements under Solution to maintain the structured approach of template design that we present.

*Structure* defines the arrangement of the component parts of the architecture. It is usually presented as a diagram.

*Dynamics* describes how the component parts relate in order to realize the *Strategy* of the architecture.

**Known uses:** Specifies existing applications of the pattern.

**Resulting context:** Describes the effects of applying the general solution to the initial context/forces, the advantages of the solution and unresolved forces. We introduce one sub element as follows.

*Adaptation/Integration* presents suggestions on how to adapt the pattern to differing projects and how to combine it with other patterns to generate agent pattern languages.

**Related Patterns:** Specifies patterns that lead to this or those that follow from the application of this or patterns that are alternates to this pattern.

See Table 1 for a summary of this template.

## 4. Examples

This section describes the BDI architecture as an example of the agent architectural pattern. Details presented are subject to space limitations.

**Name:** BDI Architecture [1, 13]

**Classification:** Agent Architectural level

**Problem:** How to build agents that need to reason in response to external events.

**Context:** Designing an architecture for agents that reason about alternative objectives and actions in a dynamic environment.

**Forces:**

*Goal*

- Goals are presented in a hierarchy.
- A choice has to be made from alternative sub goals.

Changes in the state of the environment may render a

**Table 1: Agent Architectural Pattern Template**

| Major Template Elements | Sub-elements |
|---|---|
| Name | |
| Classification | |
| Problem | |
| Context | |
| Forces | ▪ Goal<br>▪ Autonomy<br>▪ Social Ability<br>▪ Environment<br>▪ Adaptive Behaviour<br>▪ Intelligence<br>▪ Time constraint<br>▪ Decision and action |
| Solution | ▪ Control<br>▪ Knowledge Management<br>▪ Strategy<br>▪ Interaction Management<br>▪ Environmental Interface<br>▪ Structure<br>▪ Dynamics |
| Resulting context | ▪ Adaptation/Integration |
| Known Uses | |
| Related Patterns | |

▪ chosen sub goal incapable of achieving the parent goal.

*Autonomy*

▪ Timely response to changes in the environment
▪ Timely changes to commitment to sub goals
▪ Ability to handle 'malicious' agents in the environment

*Social Ability*

▪ Needs to communicate with other agents
▪ Message security is crucial to safety and success

*Environment*

▪ Environment is observable or partially observable.
▪ Environment is dynamic.
▪ Environment is nondeterministic and continuous.
▪ Environment is non-episodic.
▪ Environment could either be open or closed.

*Adaptive Behaviour*

▪ Changes in agent use.
▪ Changes in the behavioural trend of the environment.

*Intelligence*

▪ Reasoning about (large amount of) knowledge required.
▪ Goals to be achieved with optimal plans and actions.

*Decision and action*

▪ The goal and the environment are usually time critical
▪ Decisions have to be timely and accurate

**Solution:** BDI models a rational agent using the mental attitudes of beliefs, desires and intentions. Agent's knowledge is captured in beliefs such as the information about the agent's current environment. Desires represent the agent's current possible courses of actions i.e. the different goals the agent could pursue. An intention could be described as the goal that an agent commits to at a particular point in time.

*Control:* BDI attempts to balance both reactive and deliberative behaviour by having pre-determined plans to achieve goals, with triggers as to when these plans are applicable

*Knowledge Management:* The BDI model has knowledge in the form of agent beliefs and plans to achieve actions. Beliefs need to be updated in response to agent perception and changes to the environment. Dynamic changes of plans is more difficult but possible.

*Strategy:* Goal Definition Strategy: BDI architecture is based on practical reasoning which involves deliberation and means end reasoning. BDI uses the filter function and input from the knowledge bases to generate new intentions.

Action Execution Strategy: Means end reasoning is carried out by the action selection function. This function is designed to generate alternate plans when necessary.

*Interaction Management:* BDI does not specify interaction issues for the architecture

*Environmental Interface:* BDI does not specify environmental interface functions for the architecture

*Structure:* The units are structurally arranged as presented in the following figure 1.

*Dynamics:* The BDI architecture takes a perceptual input from the environment and feeds this into the belief revision function for necessary updates to the belief base. Updates to the beliefs are passed to the option generation and filter functions. The updates passed to these two functions are combined with the existing state of the intentions to update both the desires and intentions as necessary. Current intentions are fed into the action selection function to define an appropriate plan of action for achieving these intentions.

**Known Uses:** OASIS air traffic management system [11]

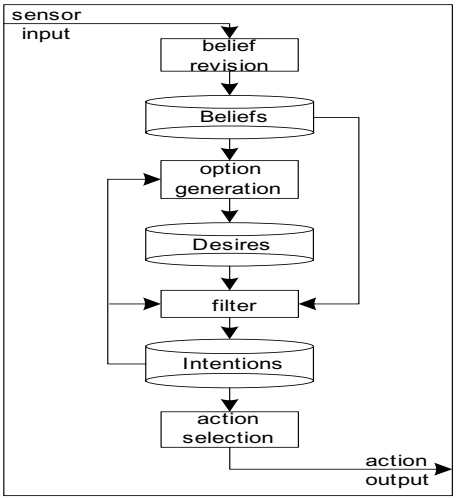


**Figure 1: The BDI architecture**

Proceedings of the 2005 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'05)
0-7695-2416-8/05 $20.00 © 2005 IEEE

**Resulting context**
Commitment to intentions: requires maintaining a balance between adequate and excessive reconsideration of an intention that the agent has committed to.
Structure: provides a good functional decomposition of the architecture into data structures and functions.
Interaction/Interface: the BDI architecture does not describe the interaction component or the interface with the environment.
*Adaptation/Integration*
The functional decomposition of the architecture makes it readily adaptable. The BDI model needs to be integrated with patterns that deal with the Control, Interaction Management and Environmental Interface components of the general agent architecture.
**Related Patterns:** InteRRaP [12]

## 5. Discussion

We discuss the pattern template presented in this paper by comparing it with two other pattern templates.

We compare the GoF [6] template and the view oriented template [8] with our template. The comparison follows:

**GoF:** The Intent and Motivation elements of the template are ambiguous and their meanings overlap [5, 7]. The template elements are not directly mapped to problem, context, forces, solution and resulting context. It also does not reflect the concepts of agency.

**View oriented:** The template elements are not directly mapped to problem, context, forces, etc. The Dependencies element introduces ambiguity since it could either describe Forces or Solution [8]. Implementation which is not required for patterns at the higher levels of abstraction is included in the general part of this template. There is no clear rationale for the choice of the view specific fields of the template.

**Our Template:** The structure is mapped to problem, context, forces, etc. The sub elements of Forces and Solution reflect the notions of agency at the right levels of abstraction. A sub element is introduced for considerations on adaptation and integration of patterns to specific projects and relevant pattern languages.

## 6. Conclusion and Further work

In this paper, we discussed the inadequacies of most of the existing agent pattern descriptions and presented a template for agent architectural patterns. The contribution of the template we presented is two-fold: One is a template structure that maintains the features of a good pattern and defines amendments to the template with sub elements. This will improve agent pattern communication, adaptation and integration. Two is a template that captures the notions of agency as integral part of template element definition. This will enhance the understanding of the agent concepts by industry practitioners.

Our further work includes defining templates for other levels of agent system development.

## 7. References


[1] Rao, A.S., & Georgeff, M.P. (1991). Modeling rational agents with a BDI-architecture. R.F. J. Allen, and E. Sandewall (Ed.), Second International Conference on Principles of Knowledge Representation and Reasoning (pp.473-484): Morgan Kaufmann.

[2] Yasuyuki Tahara, Akihiko Ohsuga, Shinichi Honiden: Agent System Development Method Based on Agent Patterns. ICSE 1999: 356-367

[3] Sylvain Sauvage, MAS Development: Reusing through Agent Oriented Design Patterns, Eight World Multi-Conference on Systemics, Cybernetics and Informatics, SCI'04, Orlando, 18-21 July 2004.

[4] Weiss, M., Patterns for Motivating an Agent-Based Approach, *Conceptual Modeling for Novel Application Domains (AOIS@ER)*, LNCS 2814, 229-240, 2003

[5] Yariv Aridor and Danny B. Lange. Agent design patterns: Elements of agent application design. In *Proceedings of the Agents-98*, 1998.

[6] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994.

[7] K. Schelfthout, T. Coninx, A. Helleboogh, T. Holvoet, E. Steegmans, and D. Weyns. Agent Implementation Patterns. Proceedings of the OOPSLA 2002 Workshop on Agent-Oriented Methodologies (Debenham, J. and Henderson-Sellers, B. and Jennings, N. and Odell, J., eds.), pp. 119-130, 2002

[8] J. Lind. Patterns in agent-oriented software engineering. online paper, 2002.

[9] R. M. Malyankar: A Pattern Template for Intelligent Agent Systems *Workshop on Agent-Based Decision Support for Managing the Internet-Enabled Supply Chain* , Agents '99,

[10] Oluyomi, A., Karunasekera, S., Sterling, L., An Agent Design Pattern Classification Scheme: Capturing the Notions of Agency in Agent Design Patterns, 11th Asia-Pacific Software Engineering Conference, Busan, Korea, December 2004

[11] Ljunberg, M., Lucas, C.A. (1992). The OASIS Air Traffic Management System. Proceedings of the 2nd International Pacific Rim Conference on Artificial Intelligence, Vol II, p1003, September 1992.

[12] K. Fischer, J. P. Müller, and M. Pischel: Unifying Control in a Layered Agent Architecture. In Proc. of the 1st International Conference on Multi-Agent Systems (ICMAS'95), p. 446, AAAI Press The MIT Press 1995.

[13] Michael J. Wooldridge. Multi-agent systems: an introduction. Wiley, Chichester, 2001.