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.


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.
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).

### 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

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

- 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](image)
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