MASIF
The OMG Mobile Agent System Interoperability Facility

Dejan Milojicic\textsuperscript{5}, Markus Breugst\textsuperscript{3}, Ingo Busse\textsuperscript{3}, John Campbell\textsuperscript{5†}, Stefan Covaci\textsuperscript{3}, Barry Friedman\textsuperscript{2‡}, Kazuya Kosaka\textsuperscript{4}, Danny Lange\textsuperscript{2††}, Kouichi Ono\textsuperscript{4}, Mitsuru Oshima\textsuperscript{4}, Cynthia Tham\textsuperscript{2+++}, Sankar Virdhagriswaran\textsuperscript{1} and Jim White\textsuperscript{2}

\textsuperscript{1}Crystaliz, \textsuperscript{2}General Magic, Inc., \textsuperscript{3}GMD Fokus, \textsuperscript{4}IBM, \textsuperscript{5}Open Group
† John Campbell is currently with the Sun Microsystems, East
‡ Barry Friedman is currently with Cisco
†† Danny Lange was with IBM during the initial part of MASIF
+++ Cynthia Tham is currently with Freegate

Abstract. MASIF is a standard for mobile agent systems which has been adopted as an OMG technology. It is an early attempt to standardize an area of industry that, even though popular in the recent past, still has not caught on. In its short history MASIF has raised interest in industry and academia. There are already a number of projects pursuing MASIF reference implementation. MASIF addresses the interfaces between agent systems, not between agent applications and the agent system. Even though the former seem to be more relevant for application developers, it is the latter that impact interoperability between different agent systems. This paper describes two sets of interfaces that constitute MASIF: \texttt{MAFAgentSystem} and \texttt{MAFFinder} (the acronym MAF is used for historical reasons). MASIF extensively addresses security. The paper provides a brief description of MASIF and its interfaces, data types and data structures.

1. Introduction

Mobile agents are a relatively new technology, but there are already a number of implementations, such as AgentTcl [6], Aglets [4], MOA [8], Grasshopper [12], and Odyssey [7]. These systems differ widely in architecture and implementation, thereby impeding interoperability, rapid proliferation of agent technology, and growth of the industry. To promote interoperability and system diversity, some aspects of mobile agent technology must be standardized. MASIF [1] is a collection of definitions and interfaces that provides an interoperable interface for mobile agent systems. It is as simple and generic as possible to allow for future advances in mobile agent systems. MASIF specifies two interfaces: \texttt{MAFAgentSystem} (for agent transfer and management) and \texttt{MAFFinder} (for naming and locating).

The original intent for MASIF was to keep it simple for the first phase and only deal with the minimal features needed for interoperability. For example, MASIF defines parameters in the agent profile to specify the requirements the agent has on the receiving agent system. This allows an agent system to support as many agent profiles as its implementation allows. Language interoperability is just one of the parameters in the agent profile. This is not a big limitation because Java is becoming the de facto standard. Therefore, interoperability in this document is not about
language interoperability. MASIF is about interoperability between agent systems written in the same language expected to go through revisions. Language interoperability for active objects that carry “continuations” around is difficult, and it is not addressed by MASIF. Furthermore, MASIF does not standardize local agent operations such as agent interpretation, serialization/deserialization, and execution. In order to address interoperability concerns, the interfaces have been defined at the agent system rather than at the agent level. MASIF standardizes:

- **Agent Management.** One can envision a system administrator managing agent systems of different types via standard operations in a standard way: create an agent, suspend it, resume, and terminate.

- **Agent Transfer.** It is desirable that agent applications can freely move among agent systems of different types, resulting in a common infrastructure, and a larger base of available system agents can visit.

- **Agent and Agent System Names.** Standardized syntax and semantics of agent and agent system names allow agent systems and agents to identify each other, as well as clients to identify agents and agent systems.

- **Agent System Type and Location Syntax.** The agent transfer cannot happen unless the agent system type can support the agent. The location syntax is standardized so that the agent systems can locate each other.

The MASIF, in its current form, provides the features required for the first level of interoperability which is the transport of agent information where the information format is standardized. Once the information is transferred from one agent system to another, how the agent system deals with the parameters internally is an implementation matter and not addressed by the MASIF standard. Such information includes agent profile which describes the language, serialization, and other requirements the agent has on the current agent system. MASIF makes it possible for an agent system to understand the requirements the agent has on its system because we believe that it is the first step in end to end interoperability.

<table>
<thead>
<tr>
<th>Function</th>
<th>Addressed by MASIF</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent management</td>
<td>yes</td>
<td>straightforward</td>
</tr>
<tr>
<td>agent tracking</td>
<td>yes</td>
<td>straightforward</td>
</tr>
<tr>
<td>agent communication</td>
<td>no</td>
<td>n/a</td>
</tr>
<tr>
<td>agent transport</td>
<td>yes</td>
<td>complex</td>
</tr>
</tbody>
</table>

Table 1 describes the types of interoperability MASIF addresses, and estimates the complexity of agent systems required to support it. **Agent management** allows agent systems to control agents of another agent system. Management is addressed by interfaces for suspending, resuming, and terminating agents. This is straightforward to implement. **Agent tracking** supports locating agents registered with MAFFinders (naming service) of different agent systems. This is also straightforward to implement. **Agent communication** is outside the scope of MASIF, and it is extensively addressed by CORBA [2]. **Agent transport** defines methods for receiving agents and fetching their classes. This requires cooperation between different agent systems, and it is complex to achieve.