A Distributed Active Information Model Enabling Distributed Autonomics in Complex Electronic Environments

Robin Braun
Center for Real Time Information Networks
University of Technology Sydney, Australia
robinb@eng.uts.edu.au

Frank Chiang
Center for Real Time Information Networks
University of Technology Sydney, Australia
frankj@eng.uts.edu.au

Abstract

Information modeling plays a central role in managing complexity of the distributed electronic systems. This paper proposes a nature-inspired distributed active information model (DAIM) to enable the local decision-making process, that will fundamentally contribute to a number of complex distributed electronic environments. The consequences from multiple distributed decision-makers give rise to the global goals exhibited by emergent properties. Details of the DAIM model are described in this paper. The validation for this model is also be given via the experimental tests in the discrete event simulator. Performance comparisons show the DAIM model outperforms the conventional information model.

1 Introduction

Current and future computing environments will be of massive scale and complexity. They have heterogeneous products, services, devices and platforms of ever-growing scale. The interactions within (or between) such systems, as well as the proliferation of standards and software from multiple vendors/operators, have led to a growing complex infrastructure which is difficult to configure, recover, maintain, operate, or re-engineer. These complex systems are characterised by unpredictability, chaotic interactions, distribution, nonlinearity and stochastic behaviour.

The conventional information model cannot survive and cannot meet the requirements from such complex, distributed electronic environments in multiple disciplines (e.g., Communication Networks including WSNs, Road Traffic SCATS System, Telemedicine and Diagnostics in health science, Water management systems and Bush fire prediction/alarming system.). An example in communications was given rise in the AC forum where John Strassner [1] challenges current stovepipe technologies used in network management, and comments that will consequently repeat stovepipe standards. Therefore it is important to rethink the OSS management structure and propose a new structure to meet the distributed self-management paradigm for autonomous communication networks (ACNs). In a word, there is a urgent need to change the current network management base structure.

Autonomic computing (AC), initially proposed by IBM, aims at exploring the high-level goal-driven self-management paradigm for these distributed environment. Viewed from the philosophical perspective, AC builds up a universal conceptual self-X framework (e.g., self-optimisation, self-configuration, self-protection and self-healing.) in order to deal with the new core challenges from heterogeneous, large-scale, uncertain and high dynamic electronic distributed environment (e.g., next-generation networking). AC is regarded as one of the promising solutions.

In view of this universal framework, the authors believe that the key factors to a well-established AC for distributed complex interactive environment are:

• Have a sustainable and maintainable information model which collects, maintains, updates and synchronises all the related information.
• Have a scalable and adaptable decision-making capability on the basis of collected information. Specifically, a scalable adaptation strategy is required for agents to autonomically manage themselves and adapt to ever-changing environments.

The research result is also applicable to communication networks of wired and wireless telecommunication networks. This research addresses an important problem in these multiple disciplines. That is, the Information Model that we have proposed, called Distributed Active Information Model (DAIM), can provide a distributed computing environment that will allow the richness of nature inspired
adaptation algorithms to be applied to such complex systems. These algorithms are intrinsically distributed, incorporating intelligent agents.

The authors have extensively described the general information models and our developments in this regard, in a number of papers [2], [3], [4], [5]. Apart from that, we have also developed nature-inspired adaptation algorithms [6] for adapting the agents and information objects for the DAIM model. This paper focuses on the details of DAIM model and how it can be practically organised and implemented.

By use of the adaptation algorithms and the DAIM model, the operation of such complex systems in any distributed electronic environment can become autonomous, scalable, interoperable and adaptable. Furthermore, the DAIM model can enable objects to make the local (distributed) decisions through its active actions (e.g., methods in classes invoked by intelligent agents). Thus, the workload of centralised decision-maker can be significantly reduced. The distributed environment requires the large number of distributed objects in the DAIM model to be highly integrated, coordinated or cooperated to achieve the system-level goal or need.

The remainder of the paper is organised as follows: Section 2 describes the relevant work in the literature about the information models. Section 3 presents the structure of the distributed active information model. As a validation test, the simulation results in Section 4 show the effectiveness of our autonomic solution - DAIM towards the handling of distributed events. Finally we conclude the contribution of this paper and future work.

2 Related Work

Information modelling (IM) has been attracting dramatic research attention currently in the autonomic communication community and operation support system (OSS) [7] society. Many research efforts on IM are closely associated with complex, distributed network environments. Information modelling provides raw information to knowledge management system for correct decision-makings. In other words, any decision or process based on the gathered information (either in real-time mode or not) can benefit from a well-developed information model.

A series of standardised information models have been currently industry standards. For example, the common information model (CIM) from the Distributed Management Task Force (DMTF) produces a set of standard approaches to mapping the information model specifically for network management based on object-oriented principles. In addition, the shared information model (SID) from the Telecommunication Management Force (TMF) establishes an integrated view from business, service, and resource domains as the standard rules for OSS designers.

The Common Information Model (CIM) standard is defined and published by the Distributed Management Task Force (DMTF). CIM presents an attempt in building up an information syntax to enable an universal understanding among different parties and therefore improve the working efficiency when all these parties have various standards. In particular, the CIM standards consist of the CIM infrastructure specification and CIM schema. The CIM schema is a conceptual schema that defines how the managed elements in an IT environment (e.g., computers or storage area networks) are represented as a common set of objects and relationships between them [8]. CIM adopts object-oriented principles to organise managed objects such as inheritance.

Both CIM and SID are seeking to develop an efficient information model to enable the self-managed functionalities in heterogeneous networks where complex electronic environments are employed. In this paper, a DAIM model with O:MIB is proposed. The comparison between O:MIB and CIM schemas and SID data models are carried out in the next section.

3 Theory

This DAIM model consists of two main parts: (1) O:MIB, (2) hybrid O:XML. The details of O:MIB is stated in this section. The details of hybrid O:XML are not repeated in this paper, they can be found in the authors’ previous papers [3], [9].

3.1 O:MIB

In addition, the pervasive issues in Telecoms Network Management (TNM) focus more on decentralisation and cooperation in contrast to traditional centralised or highly coordinated management paradigms. When the network develops quickly in size and complexity, the centralised management does not suffice. Distributed agent-based systems, as an active field of distributed artificial intelligence (DAI) in the last two decades has developed maturely. Therefore, autonomous decentralised systems (ADSs) are the best solutions so far for large-scale distributed telecommunication systems. Therefore, an object-oriented management information base (O:MIB) is needed to fulfil the service management (e.g., service discovery, service configuration, service deployment), application activation process, resource allocation process, etc. O:MIB plays a role as a distributed information model to enable an autonomous agent behavioural model which was explored with the EML model of TeleManagement Force (TMF) in our previous research framework [2], [10].

Seeking an optimal solution to achieve enormous service tasks with limited network resources is crucial in the current

\[\text{http://www.tmforum.org/}\]
network operational systems. The latest XML technology – object-oriented O:XML enables the implementation of the O:MIB to become possible, which is illustrated in the following sections. The new challenge facing us today is that the network is outgrowing the capability of existing OSSs’ propositions. For instance, on one hand, the network has more dynamics and self-configuring mechanisms such as routing, bridging, dhcp, and on the other hand, the users’ requests cannot be fulfilled in real time from the business point of view [4]. The distributed O:MIB with embedded autonomic agents (AAs) technology enables the self-organised management (e.g., resource management, service management).

The basic information unit of the O:MIB is the element or object. Each element includes:

- Attributes: specific data values that represent characteristics of the managed OIDs
- Method behaviours: actions that help to achieve autonomic communications, including the self-awareness function on time, intensity and spacial information
- Algorithms: algorithms that fulfil a particular network task can be embedded into O:MIB domains; this is a set of predefined uses of aggregations of available method calls, such as monitoring temperature, humidity in wireless sensor networks, calculating vulnerability level or raising alarm levels in ACN networks
- Messaging: messages that can be invoked by the local messaging daemon action as a response of on-demand requests in order to get general information (e.g., topology, geometrical neighbours)

This O:MIB class design conforms to the essence of distributed communication networks, and is expected to be adopted in networks such as peer-to-peer communications, mobile communication, pervasive/nomadic computing and wireless ad-hoc sensor networks (WASNs). It is envisaged that distributed agents reside on each network node, each network component (e.g., device components, service) functions with its own O:MIB. This O:MIB can be attached to the network component itself or alternatively can be in a virtual location. As Java is designed by incorporating object-oriented principles and enjoys the characteristics of being platform-independent, most of the networking applications are based on Java. The author takes Java as the programming language to implement applications. In addition, the java code can be part of the behaviours/methods/algorithms being embedded into O:MIB.

Therefore, with these methods and algorithms defined in the O:MIB, the distributed network component can calculate the important information (e.g., probability for the next traces of routing, etc.), even learning, reasoning about the situation, as well as adaptation capability. Multiple agents reside in each component as a way of activating applications when needed. Hence a full vision of autonomic communication will be achieved. The feasibility of using O:XML to describe O:MIB is further illustrated in the following section.

The necessary contents of an O:MIB class cover four divisions of information – QoS parameters, device information, service information, application information and the dependency information may be associated with services and devices. This O:MIB class is designed for each category of network components. Whenever a certain object is in need of activation during run time, the java code that creates any instance of the OMIB_Class is directed via the keyword “new”.

### 3.2 Organizing Structure of O:MIB

It is important to organise the large amount of heterogeneous O:MBs in an way in favor of the distributed environment. In order to efficiently organise and implement these O:MBs in the hierarchical telecommunication system, a distributed intelligent holonic system is adopted to accommodate them. The holonic structure is more than a component-based communication architecture but also a universal way to construct heterogeneous MEs in various levels. The holonic concepts are more likely philosophical principles which can be applied to most of the components in the universe. The holonic levels observe from the macro-system level to the micro-chip level, atomic and even nano level.

The characteristics of the proposed holonic solon agent-based O:MBs have been summarised in our paper, [11]. They are restated here again for the readers’ convenience. Firstly, they are object-oriented MIBs with methods embedded. Secondly, this O:MIB exists on a holonic-level. Holonic O:MBs are embedded into equipment, such as, individual electronic devices like mobile phone sets, printers, and even further into the sublevel of devices such as at chip-level. Applying holonic concept into O:MBs has been inspired from recent leading research in the intelligent robotics corporation (IRC) where holonic technology is embedded in individual electronic devices.

The structure of holonic O:MBs consists of three parts: (1) conventional MIB; (2) user-accessible provisioning, (3) methods/action. Each holonic subsystem is embedded with intelligent algorithms and functions to fulfil any network tasks assigned to agents and make them cooperate together, sharing synchronised information. For example, tasks such as detecting and predicting network faults and failures; QoS-constrained optimisation of network connectivity and improvement of network throughput; smart routing tasks and associated issues for wireless sensor networks, e.g., ef-
icient energy saving issues; resources (bandwidth, link capacity, services, mobile SLA agreements, etc.) and allocation issues for telecommunication networks.

Object-oriented design metrics such as inheritance and polymorphism are also applied into the hierarchical tree structure of holonic MIBs system. The base O:MIB plays the role as a superclass, whose attributes and methods are completely inherited by its subclass sub O:MIB. Holonic system concepts are applied into these O:MIB structure. Essentially, holonic systems [12] feature multi-layered hierarchies with holon entities where holonic agents reside. Each holon entity consists of multiple inherited holonic entities in the lower layer; these entities are not only associated with regard to their composition, but are also associated with their functionalities, wherein local teleholonic agents cooperate, coordinate and compete local goals so as to achieving the global goal. The shared information is retrieved by invoking the predefined methods in O:MIB rather than from a conventional information-sharing storage.

The author believes it is reasonable and beneficial to modelling O:MBIs structure into such a holarchy in the decentralised network environment. This is also the imminent requirement from the merging of distributed networks and distributed software components.

### 3.3 CIM Schemas and SID Data Models vs the Proposed O:MIB

The common information model (CIM) [13] developed by the DMTF has also provided an object-oriented scheme to organise the hierarchical data of managed elements (MEs) from different manufacturers or sources. MEs include devices and applications. With regard to their infrastructure specification and schema, the MEs are represented as classes, and the links between classes are described by associations. Most importantly, inheritance is used to describe the common base elements and inherited sub-elements. UML notation is applied and the CIM can be described in XML format in several ways. Shared information data (SID) from the Telecommunication Management Force (TMF) also has some similarities. CIM standards present a conceptual schema to encapsulate all the MEs, such as services, devices, storages, computer systems, network system and applications. The CIM essentially provides a common management information of various devices and applications, so that system administrators and management programs can access them in an universal way. The CIM schema does contain the limited methods or implementations for each class; the predefined methods are limited to static class methods. Moreover, the execution of corresponding actions are still in need of a separate high-level control (for example, centralised control) to be delivered, which is independent from CIM structures.

In contrast with SNMP MIBs, the CIM schema provides a better representation of information than static MIBs in terms of an OO structure for data-modelling. Although object-oriented efforts come from CIM schema, these efforts appear to organise the entity classes coupled with OO inheritance and associations concepts in a high level. SNMP MIBs represent information about managed objects from a low level concrete view different than CIM models. SNMP MIBs have been used in industries for many years since ISO produced this hierarchically layered model. It is known that the conventional MIB is actually a naming interface to associating with the hierarchical objects. More MIB efforts can be seen on scheduling MIBs (RFC2591) [14] and script delegation MIBs (RFC2592) years ago. RFC 2592 [15] allows the delegation of management scripts into the distributed manager for a portion of the MIBs. Although these MIB specifications explored the efforts to achieve the effect of an action being associated with a MIB variable, mostly these scripts are designed to automate tedious management processes. As such, they are not truly object-oriented.

In order to solve this problem, a new O:MIB scheme is proposed. For example, O:MIB can cope with multi-agent technology very well due to its easy implementation to the distributed bio-inspired agent structure proposed in the previous chapter. The O:MIB structure is used as a fundamental base from where other concepts and experiments are built. The O:MIB model is implemented by the object-oriented O:XML-based approach. This model can be applied to distributed autonomic communication networks to enabling autonomic functionalities. Moreover, O:XML, as a practical technology to implement O:MIB, is explored and implemented with platform-independent JAVA agents.

In comparison, O:MIB is not only providing a naming service, rather, it takes the object-oriented principle to manage the MIB objects and essentially diverts the CPU load into large amounts of local CPU. The execution of corresponding actions can be invoked by the methods and algorithms through the local agent residing on each MIB variable dynamically. Most of the decision-making tasks can be performed locally; the workload of system administrators or management programs in CIM standard can be further reduced. Therefore, the nature of the O:MIB structure is designed specifically for distributed components with local execution capabilities to improve its efficiency.

### 3.4 DAIM model

Many researchers, including ourselves, have grappled with the practicability of implementing bio-mimetic paradigms, which mimic some elements of the adaptation strategies of these colonies, in the control and management of complex systems. Most of the current literature still considers information elements as the static objects, which are...
not suitable for distributed complex environments. We consider the information objects as the active elements in this proposal for future work.

The essence of the paradigm we have developed lies in our view of the relationship between the information elements that modify the behaviours of a system, and the compute systems we use to modify them. In all current paradigms, the attributes of the information objects in O:MIB-based DAIM model are described as follows:

- Being stored in active database (termed O:MIBs in the complex systems),
- Being extracted by means of “GET” commands,
- Being operated on by some automatic or manual compute processes,
- Being returned to the store by some “SET” command.

Once these information values are in the store, they are “Bound” to some element of system behaviours. For example, the states of a traffic light or network interface. By "Active", we mean these non-static information objects are able to interact with each other, fulfilling normal maintenance functions with "store" and modify their symbols’ values during runtime by invoking their methods predefined in the O:MIB hierarchical structure.

Figure 1 shows the operations between O:XML (describing the devices or nodes details) and agents. Java agents are taken into consideration due to its platform-independent feature and moreover, most of the wide-spread agent development tools are based on Java too.

3.4.1 Use of Autonomous Agents

As depicted in Figure 1, software agents (or holon in holon-based middleware) reside on each node where O:XML is employed to Populate the recorded information to corresponding agents. The specific agents are to be instantiated from agent templates in accordance with the different electronic environments. Functions or algorithms of O:MIB can be invoked by the instantiated agents. Data values will be reconfigured by Java-based agents\(^2\). The lifecycle of the agents are fulfilled during the runtime of the program.

3.4.2 Algorithm Phases

In a brief summary, the overall stages of this approach can be described as the following steps:

1. Monitoring agents live on distributed nodes and observing the environments
2. Creating new agents when new environments are detected via the learning and adaptation strategies
3. Operating the local O:MIB by invoking the methods and algorithms preloaded into the local O:MIB systems
4. Adapting the node according to the awareness of the system-level objectives
5. Finishing off the lifecycle of the agents and waiting for next round of instantiation.

4 Experimental Validation

We present the performance analysis in this section in order to evaluate the performance of the local operations and efficiency of our proposed DAIM model. An ad-hoc network topology with six source-sink nodes is designed and implemented in J-Sim. This is an example among various electronic environments with communication aspects. As a component-based Java simulator, J-Sim is widely used to simulate the network traffic, INET protocols, wireless networks and sensor networks. In this section, we take a congestion avoidance algorithm as an validation experiment. Performance comparison is carried out between the SNMP MIB-based scheme and distributed DAIM scheme.

Figure 2 shows the topology experimented. Node_0, Node_1 and Node_2 have two or three traffic sources. Node_3, Node_4 and Node_5 are destinations for packet delivery.

To compare the performance of DAIM and standard MIB, the identical OID objects of MIB-II are also applied in the conventional MIB. Those MIB-II objects are stored in each of the node components depicted in the topology. These O:MIB variables are polled and evaluated based on

\(^2\)The reason of deploying Java agents is omitted due to the page limits.
two different schemes: one approach is based on the standards SNMP manager. The other approach is based on the proposed distributed DAIM scheme with agents.

The flowchart 3 illustrates the congestion avoidance algorithm applied with holonic agent-based O:MIB approach. By periodically observing the object ifOutOctets, the local holonic agent managers can calculate the link utilisation (LU): if the LU is greater than the threshold value \( X \), in the simulation, we set \( X = 0.6 \). When the calculation of link utilisation is above the threshold, the object ifOutDiscards is observed to ensure outgoing packets are already being discarded throughout the interface. Figure 4 shows the results when parallel events are applied into the distributed environments. The proposed DAIM scheme outperforms the standard MIB scheme. The time for discarding packet with regard to parallel events is minimised whenever the DAIM scheme is applied.

5 Conclusion and Future Trends

In this paper, an efficient object-oriented DAIM information model has been proposed and validated. The differences between O:MIB and CIM schema and the SID model have been analysed, the contribution of DAIM therefore is justified. The holonic holarchy for structuring DAIM model is also presented. The simulation results in terms of traffic avoidance scenarios show that the proposed DAIM information model and embedded smart algorithms in O:XML format outperform the standard SNMP MIB-based management structure. The future work will be employing and extending functionality of DAIM model into other real networks, such as WSNs, bush fire alarming system, etc.

References


Figure 4. Simulation Results


