A Service Based Multi-Agent System Design Tool for Modelling Integrated Manufacturing and Service Systems

Yu Han¹, Shen Zhiqi², Miao Chunyan¹, Wen Jing¹ and Yang Qizhen³,

¹ School of Computer Engineering, Nanyang Technological University, Singapore 639798
{yuha0002, ascymiao, we0002ng}@ntu.edu.sg

² School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798
zqshen@ntu.edu.sg

³ Singapore Institute of Manufacturing Technology 71 Nanyang Drive, Singapore 639798
qyang@simtech.a-star.edu.sg

Abstract

Intelligent software agent technology has wide ranging applications in automating processes that require learning and decision making. It has been demonstrated that multi-agent systems can serve the needs of a real-time distributed manufacturing environment [6]. It would be most rewarding if an integrated manufacturing and service system (IMSS) could be easily and efficiently modeled by a multi-agent system. However, although there are well established methodologies to model agent mental states, there is an absence of effective tools to convert the design into implementation. This paper proposes a service based agent system modelling and design tool – the Goal Net Designer, to bridge the gap between modeling a multi-agent system and automatically implementing the design to serve the purpose of an IMSS.

1. Introduction

Integrated Manufacturing and Services Systems (IMSS) is expected to accelerate specific areas of manufacturing, e.g. bio-manufacturing. Recently, there is a need to extend high throughput life science research to bio-manufacturing. Service oriented architecture (SOA) is a convergent strategy and trend for IMSS. Within SOA, services can be composed and reused in workflows and business processes. Therefore the whole manufacturing system is integrated by workflows and business applications. However, business process modelling and management become critical in such system as business processes keep changing all the time. So a tool for dynamically modelling a business process based on available services is needed. In this paper, we introduce a goal oriented approach for process modelling and present a service based process modelling tool developed in our research. The process modelling tool is also an agent modelling tool so that a created process model can be executed by agents automatically.

The goal oriented model most closely resembles how people, in the day to day activities, make decisions to direct their actions. It has great potential to be understood easily by people with little expertise in intelligent software agent technology. Nowadays, most of the available intelligent agent design tools require the users to be experts or at least highly skilled in the field of intelligent agent to design a multi-agent system or agent mental states and even more highly competent in programming to implement the design. Therefore, one aim of this research is to provide a tool – the Goal Net Designer, for less technically inclined people (manufacturing process planners etc.) to enable them to actively participate in the design of multi-agent systems.

The working environment for end users is a service-based web portal which facilitates collaboration and sharing among users. Manufacturing services, workflows, complex computing processes and modelled business processes designed using the Goal Net Designer are shared among the users and updated iteratively. The Goal Net Designer is fully integrated into the portal so that users can create and execute new business process online using the available services, workflows and other processes.

2. Related Work

Currently, there are several design approaches for the design of multi-agent systems and agent mental states. One of the most popular tools for intelligent agent development is the Java Agent DEvelopment Framework (JADE).
However, it is more of an Application Programming Interface (API) [2] than a tool for conceptual design. In order to create an agent in JADE, one has to be familiar with both the Java programming language as well as the agent oriented programming paradigm. This is more often than not too much to expect from a designer of a game plot or storyline of an agent augmented interactive digital media application.

Another approach of designing multi-agent system and agent mental state is to sketch the design on paper and feed it to a computer with software capable of making sense of it. In most cases, the software is platform and language dependent. Although the SketchiXML project offers a solution to this problem [3], it has nevertheless some inherent drawback by using this approach. Firstly, users must draw out the design following closely to the shapes of the recognized entities involved. Secondly, as the design is being modified, the user must also follow certain ways to delete unwanted items or simply start all over again. In the end, this approach does not enable the users to specify internal properties of each design item during the sketching phase. The skill level required to make use of this design approach is high and the implementation is still a long way to go from the design.

The Prometheus Design Tool (PDT) proposed by RMIT University, Australia, provides support for the various stages of multi-agent system design using the Prometheus methodology. It has been integrated with Eclipse to provide a more closely coupled implementation process [4]. However, it still requires the users to have profound knowledge in a particular agent modeling methodology – Prometheus and the ability to write programs based on the design. The data acquired at design time serves only as a reference to the implementation. The agents based on the design can not be created automatically from the design data.

3. Overview of the Goal Net

A Goal Net is a composite goal hierarchy which consists of goals/states and transitions [1]. In detail:

- A goal net is a hierarchical net. It contains a set of goals/states, transitions and arcs organized into a network to achieve an overall goal.
- A state is a tuple containing a set of variables that define the profile of the state, a set of application variables and a set of internal functions.
- A composite state is a state which contains additional information on the initial and target states of the sub-goal-net it represents.
- A transition is tuple containing a set of variables that define the profile of the transition, a set of application specific variables and a tasklist which consists of a finite set of tasks each, in turn, consists of a finite set of functions.
- An arc is a tuple containing a set of variables which define the profile of the arc, a link to the input state/transition and a link to the output transition/state.

The goals/states are used to represent the goals that an agent needs to go through in order to achieve its final goal. The transitions connect one goal to another specifying the relationship between goals it joins. Each transition must have at least one input goal and one output goal. Each transition is associated with a task list which defines the possible tasks that agent may need to perform in order to transit from the input goal to the output goal. A complex system can be recursively decomposed into sub-goals and sub-goal-nets. In such a manner, the system can be easily modeled and simplified.

4. Multi-Agent Development Environment

The goal oriented agent design methodology which is based on the Goal Net model covers the entire life cycle of the agent system development from requirement analysis, architectural design to detailed design and implementation [1]. It has a practical implementation called the Multi-Agent Development Environment (MADE) which is divided into the Goal Net Designer and the MADE Runtime. The MADE system elicits agent mental state design from the users via a friendly user interface and generates data for MADE Runtime to create agents based on the design and manage them.

![Figure 1. MADE System Architecture](image-url)
As shown in Figure 1, the Goal Net Designer supports goal and transition design, environmental variable specification and, as it is implemented as a client-server system, collaborative development among multiple users. With an easy to use interface based mainly on drag-and-drop operations, users can modify the design in an iterative manner with less effort using the Goal Net Designer. This is especially important because changes to part of the goal net may affect other parts and it is error prone and difficult to manually keep track of them. Functions and Tasks are important concepts in the Goal Net methodology. They are essentially actions and decision making logics that the agent created based on a particular goal net should perform at its various states and transitions. Many a time, a function or a task is generic enough to be used in more than one goal net. The goal net designer supports function and task management to enable the users to reuse them in multiple development projects.

The MADE Runtime, on the other hand, makes use of the data acquired via the Goal Net Designer in the Goal Net Database. It creates agents based on the goal net designs from the users and manages the agents by executing the transition tasks, modifying and keeping track of environment variables and managing agent communications. It is the MADE Runtime that actually converts the users’ designs into implementation.

5. Goal Net Designer

The main purpose of the Goal Net Designer is to make the detailed formatting and storage of goal net data transparent from the users so that they can concentrate on designing the logics of the goal net. As the design is being modified, the underlying system generates data structures using the information provided by the users in real time so that when the design is finalized, the entire goal net is generated in memory and stored in a format which can be easily used by MADE to create the agent and run it.

The Goal Net Designer provides a simple to learn and
easy to use drawing canvas for the users to design the goal net with as little effort spent on drawing and layout as possible. Thus, the entities involved in a goal net – atomic state, composite state, simple transition, reasoning transition, directed arcs and parallel arcs etc, are all provided as standard components which can be dragged and dropped and moved about on the canvas.

5.1. Function and Task Reusability

In a goal net, each state may be connected to more than one transition. Therefore, there may be times when decisions have to be made as to which path to take from the list of choices in the current state. MADE provides default action selection methods based on generic algorithms such as shortest path first, but users can also define their own criteria. Therefore, each state has a list of functions attached to it which may consist of several different algorithms to select the next transition.

A transition of a goal net is essentially a sequence of actions an agent has to perform in order to move from one state to another. Therefore, each transition has a list of tasks which, in turn, consists of a sequence of functions. As some tasks are frequently used in more than one goal net, instead of creating them over and over again, the goal net designer retrieves all tasks which have been created from the database and lists them out on an expandable tree view as shown in Figure 2. Then, users can again apply drag-and-drop operation to associate transitions with tasks.

![Figure 3. Functions from DLL Files and Web Services](image)

Figure 3. Functions from DLL Files and Web Services

A set of core functions compiled into dynamic linked libraries (DLL) resides on the server. Their details (including function signature, description and alias) are loaded into the central database. Each time a client connects to the server, the function information is automatically passed to the client to be displayed in an expandable tree view form as shown in Figure 3. Upon finding a useful function for a state, the user may drag the function and drop it on the desired state.

Users can also choose to use web services either provided by a third party or implemented by themselves. They add a new web service at the goal net designer through the grid-based development portal by providing the necessary details of the services and the portal will generate necessary files to make them available for use to all users in the future.

5.2. Ensuring Consistency

As a goal net design is being modified, existing entities may be deleted. However, as the design is a network with hierarchical structure, a deletion may have ripple effect which affects other parts of the network depending on the location of the deleted entity in the entire hierarchy.

For example, if a state at the start of a sub goal net is deleted, all the outgoing arcs connecting it to the next transitions need to be deleted, the input arcs records in all the transitions originally connected to the deleted state have to be updated and the start goal record at the composite state representing the sub goal net needs to be updated to reflect the change. Moreover, with a state missing in the goal net, the design can not work correctly during implementation. Thus, the user has to be reminded of the fact that this design is not workable and further modifications are necessary. To manually keep track of all these changes can be tedious and error prone as the design becomes more and more complex.

The goal net designer automatically takes care of all the ripple effect created by adding or deleting entities in a goal net design. At the end of the design session when the user chooses to save, the goal net designer checks the design against the rules stipulated by the Goal Net Methodology and alerts the user if the components critical to the implementation phase are missing. This way, the users can have a better idea if their design can work or where the errors may be before committing it to implementation.

5.3. The Goal Net Designer on the Web

The main process of developing a multi-agent system to assist IMSS is carried out on the manufacturing portal where users can find all available manufacturing services, workflows, computing processes and business processes modelled using the Goal Net Designer. In a business process, a web service, a workflow or another process is invoked in a transition as a task of the transition. A user
6. Goal Net Designer in Action

Having discussed the features of the goal net designer, in this section, we will further illustrate the application of it in various stages of an agent design. Since this design tool targets multi-agent augmented integrated manufacturing and service systems, the process of designing an agent using the Goal Net Designer is also representative of specifying a business process or a workflow in an IMSS.

6.1. Requirement Analysis

Requirement analysis is the initial stage in many software engineering methodologies. In the Goal Net methodology, goals are of critical importance in the elicitation and elaboration of the requirements. To better represent the objectives of an agent, a set of goals (what are to be achieved) are first organized into a hierarchy and the transitions (how to go about achieving the goals) are ignored for the moment. Complex goals can be further decomposed into sub goal nets within the overall design and the overall object can be illustrated in a top-down approach.

The goal net designer offers a visual solution to this development stage. It allows goals (states) to be added to the design without considering other entities. Users can organize the goals in a visual pattern such that the hierarchical structure is the most intuitive to their eyes. As the design evolves, some atomic states may need to be further decomposed. With the help of the goal net designer, all necessary steps to convert an atomic state to a composite state and vice versa are made transparent to the users.

At this stage, the hierarchical organization of the goal net has not actually been set up. What the users see is a visual layout pattern that makes sense only to the human eyes. Therefore, conceptual hierarchy can be altered by simply moving a state to a different location on the canvas. At this moment, the design makes sense only to the designers.

Once the goals are clearly defined, the way to achieve these goals has to be added into the design. This step is realized by using the transition entities in the Goal Net methodology. Different transition each specializes in performing a particular task or set of tasks can be listed out in the goal net designer canvas. Whenever a transition is deemed suitable for achieving the next goal based on the current goal, it can be connected from the current state to the desired next state using arcs. One transition can depend on more than one state to be fired and it can result in more than one next state being reached.

As transitions are connected into the goal net design, the goal net designer gradually builds up the hierarchical architecture in the background without the users’ being aware of it. Once the design is finalized, the hierarchy can be put through a series of checking to ensure its correctness based on the rules set out by the Goal Net methodology.

6.2. Detailed Design

After the requirement analysis, the skeletons of the goal net design have been drawn out. To make the design workable, details in each state and transition have to be filled in.

The MADE provides default goal selection algorithms which depend on the properties of each state. The basic ones include the cost and achievement of reaching that goal. Moreover, users can define custom properties at each state to be used for their own goal selection algorithms. The goal net designer provides the utility for the users to specify the name, type and optionally the default value of the custom defined properties and the function facility mentioned in earlier sections helps the users to associate functions defined elsewhere to the states.

A transition is specifies how an agent created based on the goal net design moves from one state to the next. Therefore, it is mandatory for it to have at least one task which contains at least one function. The task facility discussed in earlier sections offers an easy way to the users to manage and reuse the tasks for each transition.

Once the detailed design stage is finished, the goal net contains enough information to be used by MADE to automatically generate agents and execute the model.
6.3. Case Study

In our research, we proposed a service oriented architecture for IMSS in a grid based environment. Each grid hosts a group of manufacturing services. We developed a community portal for users to form a community to share and utilize services in this portal. Users can be service provider who publishes services through this portal or service consumer who creates business processes on the portal using the Goal Net Designer.

A bio-manufacturing prototype system was developed in this environment. This prototype system aims to prove the concept of our solution to IMSS and demonstrate how bio-workflow is integrated over grid computing and how grid-enabled bio-instrumentation as available services are used to design and fabricate prototype DNA chips for rapid development of diagnostics in the event of a major outbreak of new strains of infectious diseases.

In this system, we first wrapped available bio-functions as web services and published them on our portal. Then we published the available bio-workflows on the portal as well. Then we invoked the Goal Net Designer to create a simple process using the published web services and workflows. The simple process model was shown in Figure 2. After we designed the process model, the model was saved in the database on the server and a link was created in the business process portlet. The user can modify the model using this link or execute this process through this link. When we executed this process, agents were created and the process model was loaded into the agents. The agents executed the tasks according to the model. The results were generated as expected.

The advantages of the proposed tool include:
- Business process can be modelled on demand online using up-to-date web services, workflows and other available functions and processes.
- The tool can be invoked through web browser.
- It supports drag and drop method for the design.
- The created model is a goal net model which is agent model. So the process can be easily executed by an agent system.
- It provides a real platform for IMSS.
- It enables collaboration through web based community portal among users in IMSS.

7. Conclusions and Future Work

With the Goal Net Designer and MADE Runtime, users can be freed of the concern of implementing their agent designs which is a process normally requiring in-depth knowledge of the intelligent agent and programming. Thus, system planners can focus their attention on the detailed design of each business process and the complex system operations of an IMSS instead of worrying about how to convert their designs into implementations. The time and cost of developing an agent augmented integrated manufacturing and service system can be greatly reduced.

In the further development of the goal net designer, the agent architectural design phase of the Goal Net methodology which concerns the reusing of goal nets to form a multi-agent system and the issue of agent communication and management will be catered for. A more comprehensive validation module to provide detailed diagnostic information for the designers will also be incorporated into the Goal Net Designer. The Goal Net Designer interface will be further defined so that more functions and features can be provided to users to facilitate more convenient development of agent augmented IMSS.

Acknowledgements

This work was supported in part by the Agency for Science, Technology, and Research (A*STAR) of Singapore under the Thematic Strategic Research Programme: Integrated Manufacturing and Service Systems (IMSS) Project 0521160070.

References

[3] Adrien Coyette, Stéphane Faulkner, Manuel Kolp, Quentin Limbourg, SketchiXML: towards a multi-agent design tool for sketching user interfaces based on USIXML, Jean Vanderdonckt Université Catholique de Louvain, School of Management (IAG)