行政院國家科學委員會補助專題研究計畫成果報告

※※※※※※※※※※※※※※※※※※※※※※※※※
※　　　　　　　　　　　　　　　　　　　　　　　　※
※　**智慧型雙向傳呼之應用產生與資料庫連結之研究**　※
※　　　　　　　　　　　　　　　　　　　　　　　　※
※※※※※※※※※※※※※※※※※※※※※※※※※

計畫類別：√ **個別型計畫**　　□**整合型計畫**
計畫編號：NSC　89ー2213ーEー032ー007ー
執行期間：　88 **年**　8 **月**　1 **日至**　89 **年**　7 **月**　31 **日**

計畫主持人：劉艾華
共同主持人：

本成果報告包括以下應繳交之附件：
□**赴國外出差或研習心得報告一份**
□**赴大陸地區出差或研習心得報告一份**
□**出席國際學術會議心得報告及發表之論文各一份**
□**國際合作研究計畫國外研究報告書一份**

執行單位：淡江大學資訊管理系

[1] **中　華　民　國**　90　**年**　2　**月**　1　**日**

行政院國家科學委員會專題研究計畫成果報告

使用階層式知識庫機制建構智慧型網路應用架構

Constructing Intelligent Network Application Framework with Hierarchical Knowledge Base Mechanism

計畫編號：NSC 89-2213-E-032-007
執行期限：88 年 8 月 1 日至 89 年 7 月 31 日
主持人：劉艾華　　淡江大學資訊管理系

中文摘要

雖然網路科技發展速度驚人，但是仍有空間發展出更先進的應用與更方便的服務。本研究建立先進的智慧型多代理員架構，有多種型態的服務代理員共同為網際網路上的使用者服務。為了在代理員間形成一致性的工作觀點，本研究提出一個具彈性的知識架構以模擬所處理的問題。為了幫助服務提供者產生過渡性的使用者介面，本研究亦提出一種 Knowledge Query Interface Script (KQIS)以動態的在使用者端產生介面。

關鍵詞：智慧型代理員，知識分享，資料庫應用，使用者介面

Abstract

Although the networking techniques are developing amazingly fast, there are still rooms for more advanced applications and more convenient services to be discovered. We have established an advanced intelligent multi-agent framework with various types of service agents cooperating and serving users on the Internet. For forming consistent business perspectives among agents, we propose a flexible knowledge hierarchy to model the problems at hand. To assist the service provider produce transient user interfaces for interacting with user, we also propose a Knowledge Query Interface Script (KQIS), to dynamically generate user interfaces on client. This research adopts several applied techniques, like KQML and Jess.

Keywords: Intelligent Agent, Knowledge Sharing, Database Applications, User Interfaces

1. Introduction

Background

This research discusses a framework, which allows a group of cooperating intelligent agents interacting with each other, these agents use a shared ontology mechanism for dynamically enhancing capacity and competence. With more and more popular developing of network services, it is desirable to have an intelligent, friendly, and powerful way to manage, classify, collect, and publish organizational data and related occurring data. It is also appealing for techniques of constructing services that could be combined with integrated agent technologies.

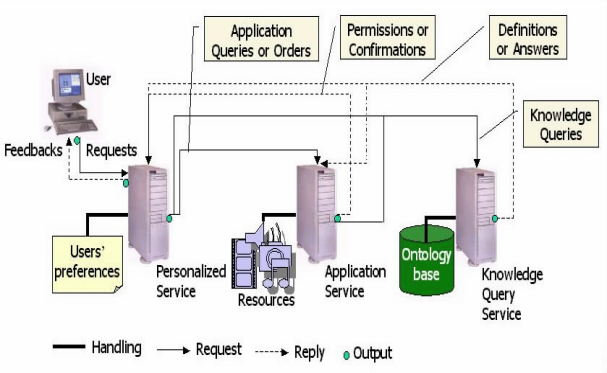
However, before our goals can be properly achieved, there are several issues should be addressed. First, the data definitions of an entity between different applications are usually inconsistent. It causes heterogeneous data problems with inconsistent field names and data types. Second, The diversity and un-integration of protocols between applications limits and disables the development of multi-agent environment. Third, extant multi-agent technologies are rarely considered with developing reusable and shared ontology base, it causes the inconvenience of communication among agents and restricts the consensus of agents with the knowledge terms.

Research Motivations and Objectives

This research attempts to wrap conventional applications with intelligent agent interface for assisting users and facilitating business operations. For integrating data schemas and knowledge terms, this research has constructed an expandable ontology base, and implemented a practical conceptualization model to provide a solution to define each pattern which is used in a bounded domain application. |While developing such an application, all the term have a prepared and reusable definition, and the developer need less effort on defining and considering the consistency of each entity in the application.

2. System Framework

In order to be clear about these notions, this research distinguishes the works into personalized service, application service, and knowledge query service. (Figure 2.1) Personalized service unit interacts with users



directly to capture users' requests, maintains users' profiles, and act for user to submit

Figure 2.1: System framework

requests. Application service unit handles enterprise or household resources, such as routine databases. Knowledge query service unit is responsible for maintaining an ontology base in an organization and providing the related knowledge definition of processing tasks, which accepts knowledge queries from other units and replies the indications or procedure lists about the task.

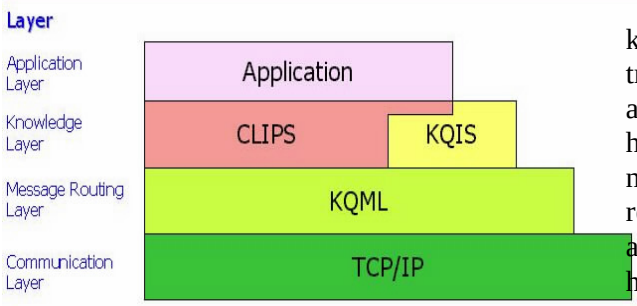


Figure 2.2: Layered Architecture

From vertical viewpoint, the message in the environment represented as layered architecture shown in Figure 2.2. Since the communication among each unit is streaming over the Internet, TCP/IP is the basic communication layer. The next layer, message routing layer which provides addressing mechanism across the distributed environment among each service unit. The third layer is knowledge layer, which has two parts: CLIPS and KQIS. CLIPS layer is used to represent knowledge and process reasoning. KQIS layer provides a dynamic query interface between knowledge and user interface. The top layer operates business rules and executes service functions.

Knowledge Hierarchy

The idea of knowledge hierarchy was triggering from domain name service (DNS). With the concept, this research defines and addresses all terms in a domain application as knowledge definitions, and associates those definitions with a tree structure. Every service unit, agent, holds and maintains its own knowledge hierarchy as an exclusive knowledge base. Every path of knowledge hierarchy defines the general name, derived-from definition, access mode, and the containing properties. As DNS, there is a master in each knowledge area and all definitions are originated from the root: "$null".

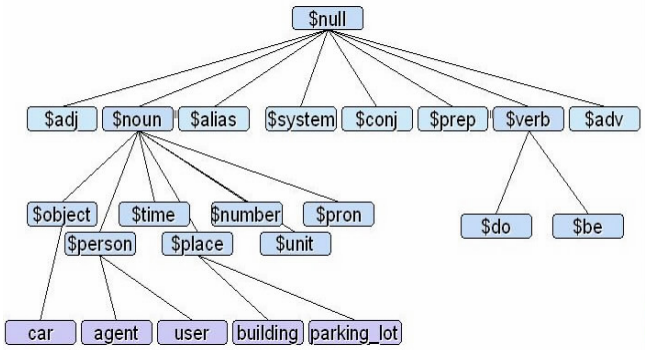


Figure 2.3: Knowledge hierarchy

As Figure 2.3 presented, there is a knowledge hierarchy consisted from a group of tree nodes level by level. This research uses the architecture to organize the concepts that human have recognized in object-based manner. Each node can be assigned a particular concept and the related attributes that are concerned by human are recorded. If a concept is to be placed into the hierarchy, it can be formatted to include the *name*, the *derived-from category*, the *access modes* and the *attributes*. The name is the representation of the concept. The derived-from category is used to point out the classification of the concepts that have common access modes and attributes. The access modes are to specify the methods of how to control or access the concept. The attributes hold the values that instance has occurred. As the top-level node, it is the original concept on the hierarchy, which indicates the *null concept* and keeps the basic attributes. All concepts are originated from the null concept and to be developed for the specified purpose on the architecture. This research has established a simple view of the environment based on the knowledge hierarchy.

Knowledge Query Interface Script

The Knowledge Query Interface Script (KQIS) is a protocol, which is used for coordinating knowledge expressions and user interface. When personalized service agents

need to confirm or ask users conditions, they will generate KQIS scripts autonomously to client-side, and client-side interface generator forms the user interface. Presently, KQIS has eight types of formats for arranging components on user interface. A script has a tag name to capture external data from the component and correspond to the slots of CLIPS templates. Finally the response scripts will be translated to standard CLIPS expressions on server-side.

3. System Implementation

This research has implemented the framework into the "Frontagent" package in JDK 1.3 with inference engine package: Jess 5.0. The most essential utility of the package is to provide the agent interface and capability for wrapping domain applications without bothering users about how to deliver the details of the services.

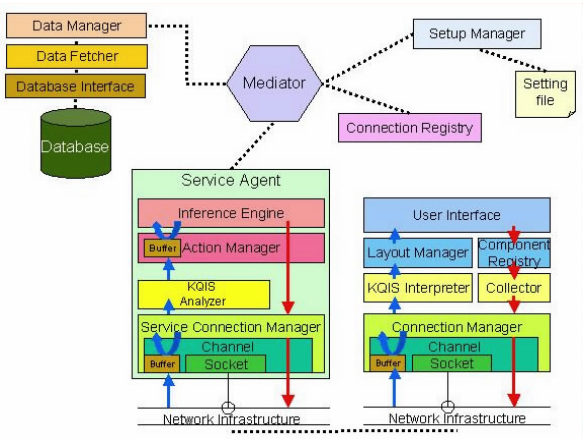


Figure 3.1: System architecture

As Figure 3.1, the framework can be defined as 4-tiers client-server architecture. The outmost front is the user-side, which handles interface presentation but without processing the business rules. The server-side produces services and supports connection maintaining, messages routing, configurations setting, and ontology querying. This research distributes the back-end procedures into three services: personalized service, application service and knowledge query service, and assigns three actors: user, knowledge engineer, and the service agent collaborating with each other to achieve these goals in the operating environment. Users have the authority to assign a new task and review historical tasks. A knowledge engineer is responsible for managing the knowledge definitions of the ontology base set up on knowledge hierarchy, includes modifying for the domain demand and maintain it periodically. Service agents are acting on the server. A service agent can communicate with others, but it doesn't handle the networking mechanism directly. It delegates the server to establish a connection to other site, and then asks server to forward messages to other agents.

4. Example Application

In this section, we explain an example application for parking lot searching. Assumes that, a software company manager named Johnny, his personalized service agent, Jacky, always monitoring and assisting the arranging of Johnny's schedule in the agent server. (Figure 4.1)

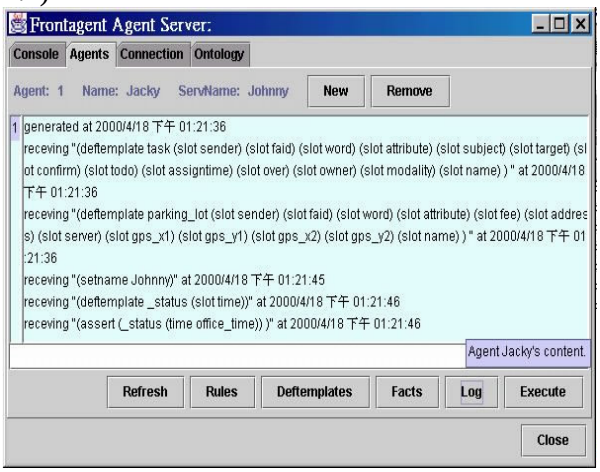


Figure 4.1: Frontagent Agent Server Interface

One day, Johnny has a meeting in Taipei world trade building at PM 1:00. Jacky detects the schedule item and starts the background works. First, Jacky analyzes the schedule described in CLIPS. Then, Jacky gets "Taipei world trade building" from "location" slot and asks Johnny if he needs to order a parking space near the destination. Then, Jack loads the parking lot order procedure to perform the ordering mission. After a series of interacting and reasoning, Jacky gets a list about the parking lot and shows it on the user interface (Figure 4.2).

Figure 4.2: user interface with list about parking lots

5. Conclusion and Future works

This research has presented a cooperating multi-agent framework that covers the topics of knowledge representation, knowledge querying, ontology integrating, task-oriented reasoning, and communication mechanism. The knowledge hierarchy is responsible for constructing knowledge definitions and consensus associations, which allow agents to have a common observation of their businesses and tasks. It is also a protocol for integrating the shared ontology bases. The CLIPS and Jess is used to shape the intelligence for knowledge representation and agent reasoning. We also applied KQML for establishing the handshaking message routing communication. We can use the framework to improve conventional services and provide a more convenient and efficient environment. In present works, we are short of a simple interface to assist service providers to construct their services and maintain their business rules. We also haven't considered the framework with agent mobility and location synchronicity, and there may be conflicts existed between knowledge definitions and ontologies. It is possible, in the future, to except for expanding the work to allow agents mobility, reinforce current knowledge flowing mechanism, enhance knowledge plug-and-playing, extend client-end structure with servlets on web-based model, and expand KQIS with some defined functional dialogs and Java 3D API.

6. Self Evaluation

Since the technical environment in Taiwan without the practical incentive for two-way paging, this research has adjusted to the applications connected mainly via Internet. Since the connecting protocol can be adopted both way, it can be transformed to two-way paging immediately if the infrastructure for two-way paging emerged and become accessable. Therefore, this adjustment do not obstacle this research in terms of its required effort or possible achievement. This research result has been submitted to the conference of the 12th International Conference of Information Management.

Reference

[1] Farquhar, A., R. Fikes, J. Rice, "The Ontolingua Server: A Tool for Collaborative Ontology Construction." Knowledge Systems Laboratory, KSL-96-26, September 1996.

[2] Finin, Tim, Jay Weber, "DRAFT Specification of the KQML Agent-Communication Language", The DARPA Knowledge Sharing Initiative External Interfaces Working Group.

[3] Friedman-Hill, Ernest J, "Jess, The Java Expert System Shell", http://herzberg.ca.sanda.gov./jess/

[4] Gruber, T.R, "Ontolingua: A mechanism to Support Protable Ontology", KSL, Standford University, Knowledge systems Laboratory, 1992, pp91-99.

[5] Haugeneder, H., D. Steiner, "Co-operating Agents: Concepts and Applications.", Agent Technology Foundations, Applications, and Markets, pp176-179.

[6] Indermaur. Kurt, "At Your Service? A State-of-the-Industry Report on Electronic Agents for E-Commerce.", DBMS, Sep 1998, pp31-36 & 53.

[7] Janca, P.C., D. Gilbert, "Practical Design of Intelligent Agent Systems", Agent Technology Foundations, Applications, and Markets, pp75-76.

[8] Janca, P.C., D. Gilbert, "Practical Design of Intelligent Agent Systems", Agent Technology Foundations, Applications, and Markets, pp75.

[9] Maes, Pattie, "Agent that Reduce Work and Information Overload", Communication of the ACM, 37(7), pp31-40.

[10] Mayfield, James, Yannis Labrou, Tim Finin, "Evaluation of KQML as an Agent Communication Language", Intelligent Agent Volume II, 1995.

[11] Reley, Gray, "A Tool for Building Expert Systems", http:// www.ghg.net/clips/CLIPS.html, Aug 1999.