

智慧型電腦代理人之設計與研究

The Design and Implementation of Intelligent Desktop Agent

Cheng-Lung Sung 宋政隆
Wen-Lian Hsu 許聞廉
Shih-Hung Wu 吳世弘
中央研究院 資訊科學研究所
台北市南港區研究院路 2 段 128 號
E-mail: {clsung,hsu,shwu}@iis.sinica.edu.tw

Wei-Kuan Shih 石維寬
國立清華大學 資訊工程學系
新竹市光復路二段 101 號
E-mail: wshih@cs.nthu.edu.tw

摘要

本文提出了一個以自然語言理解 (Natural Language Understanding) 為基礎之智慧型電腦代理人 (Intelligent Desktop Agent, IDA) 架構。使用者使用自然語言描述需求, 代理人理解之後, 會判斷出相對應的動作及指令。智慧型電腦代理人也能代替使用者執行大部份的動作與指令, 讓使用者可以使用自然語言完成他想要達成的工作。

我們藉由智慧型的人機界面提供一個能理解並處理自然語言指令的代理人, 以多代理人系統 (multi-agent system) 的方式運作。接受到指令之後能以任務的不同, 分派給其他代理人。透過多代理人彼此的溝通合作, 輔助使用者做有效率的電腦操作; 另一方面透過系統實作, 結合自然語言理解與代理人程式的技術, 有效應用在電腦操作問題上。

關鍵詞: 代理人(agent), 自然語言理解(natural language understanding), 多代理人系統(multi-agent system)。

1. 導論

一般人接觸電腦的時候, 經常遇到對於作業系統環境不熟悉的問題。使用者雖然知道自己想要達到的目的, 卻不清楚所需要操作的步驟及指令。即使以圖形化使用者界面(Graphic User Interface) 為主的視窗作業系統, 對於較少使用的命令依然會有操作上的困難。在一些以命令列模式為主的作業系統如 Linux, FreeBSD 等等問題更嚴重。即使是有經驗的使用者, 在作業系統更新版本使得操作方法改變時, 還是要面臨重新學習如何下指令的問題。

本研究中, 我們應用代理人技術來解決上述問題。觀察過去的研究, 電腦代理人 (Desktop Agent) 被證實可以自動完成一些工作, 以提升使用者的工作效率, 並減輕其負擔[5] [10]。而界面代理人 (Interface Agent) 則能接受使用者的輸入,

並以可見的物件與使用者做互動, 引導使用者完成工作[5] [9]。例如微軟在其 Office 系列軟體中加入了小幫手輔助使用者解決問題, 並且可與使用者互動。微軟的界面代理人程式也可以接受語音輸入指令, 從而節省使用鍵盤及滑鼠的時間[11]。大多數的電腦系統都有提供一個有幫助的輔助工具, 可以提供使用者做文字查詢並提供線上的求助。Gwei 與 Foxley 曾經研究以加入使用者特殊的需要或是經驗做為對照說明解釋的方法來改善線上求助系統[4]。但是大部份的系統並沒有足夠的知識基礎來理解使用者的需要, 使用者必須給出完全正確的指令, 而不能只是簡單地描述要完成的工作。

以滿足使用者的需求為主要設計目標的智慧型代理人, 在實作上是理解使用者的需求為優先, 再根據所分析出的資料對代理人下指令[12]。使用者將如何與代理人互動是一個有趣的論題。一般性的作法是以取出關鍵字的方式設定某些動作及目的來達成代理人的應用, 在電子郵件軟體的操作上更可以萃取出有用的資訊[1]。自然語言在 Internet 商業上的應用可以幫助使用者以互動式的交談來逐漸縮小搜尋資料的範圍, 最後得出最接近的產品資料[2]。

因此, 我們發展一種電腦代理人機制來幫助使用者。當代理人接受到使用者對此次操作主題的自然語言描述, 理解之後可以取代使用者完成中間的操作過程, 達成使用者的要求。例如, 設定電腦時鐘, 設定網路連線等等。除此之外, 如果指令是為了視窗操作, 檔案搜尋, 網際網路資訊等等不同類型的即時資訊取得, 則可以運用多代理人的理論來整合。針對代理人的責任與任務, 以資訊取得方式來區分為網際資訊代理人, 檔案管理代理人, 執行代理人等。經由類似 KQML (Knowledge Query and Manipulation Language)[3] 的代理人通訊語言 ACL (Agent Communication Language) 可以達成不同代理人的溝通與合作, 可以達成一些更複雜的任務, 例如可以由檔案管理代理人取得昨日最新增加的文件檔, 並由執行代理人來做開啟編輯的動作。

2. IDA 系統

智慧型電腦代理人以理解使用者輸入自然語言的需求，幫助使用者執行相關的指令及操作為主要目的。

2.1 IDA 系統知識來源

使用自然語言做為輸入命令的語言方式減低了使用者的負擔，但是系統必須要具備大量範圍的知識。微軟視窗作業系統 (Microsoft Windows) 的線上說明手冊 (On-line Help) 的逐步 (step-by-step) 教學機制可以做為一個由單一查詢或指令延伸到一系列的指令的知識來源。將自然語言指令對應到輔助說明文件，再找出需要執行的電腦指令，是以資訊資料庫知識概念[6] [7] [13] 為基礎。在這個結構下，代理人只需要有基本的系統知識便可以完成許多複雜的任務。

2.2 InfomationMap 系統概說

中文自然語言的語意理解是以許聞廉教授領導的智慧型代理人系統實驗室發展出來的 Information Map (InfoMap) 做為核心[8]。而 InfoMap 的一個應用就是「中央研究院智慧型查詢系統」。智慧型查詢系統 (QA System) 是理解使用者以自然語言提出的問題，找到與知識庫內整理過的最接近的 FAQ，將答案提供給使用者。由於中文字詞的意義並不像英文每個字之間都會有空白隔開，而是以句子為一段，因此在處理字詞分析上比起英文來說難度較高。透過 InfoMap 處理中文，可以正確地分析出中文的字詞意義，進而可以處理使用者的中文自然語言查詢。如問句「請問中研院是在那裡呢？」，InfoMap 將會分析出 **中研院**、**在那** 的關鍵字，並對應到 **中研院的地點** 這個 FAQ。

IDA 利用 InfoMap 處理中文自然語言的能力，增加了自動處理指令的能力。原有的 InfoMap 只能提供出靜態的資料，應用在個人電腦上則只能做為一個 FAQ 的系統，而 IDA 更能加強原有功能，幫助使用者處理後續的操作。

2.3 IDA 系統說明

使用者的需求可以分為不同的操作主題，單一的操作主題描述著一個事件，而這一事件包含了一個或一個以上的操作指令。我們只要能對應使用者的自然語言指令到正確的操作主題，接下來的步驟就只要對於主題內的事件動作來做處理即可。

以微軟視窗作業系統的線上說明手冊的搜尋

功能來做實驗。微軟視窗作業系統說明手冊原有的搜尋功能只能接受關鍵字查詢，因此使用者所輸入的自然語言指令若沒有用到正確的關鍵字將無法查詢到適當的資料。舉例來說，如果我們在說明手冊上輸入的字串是「我想要改變螢幕解析度」而不是「變更螢幕解析度」，那說明手冊將會出現「沒有找到主題」的訊息。另一個常見的問題則是，即使輸入的關鍵字對應到正確的文件，一些不熟悉操作的使用者在看到說明手冊上的「開啟 **控制台**」的敘述，也會有找不到或是不知道 **控制台** 在那裡的窘境。

相對地，IDA 會分析使用者的需求，對出現的字詞做近意語的擴張，然後挑選出有意義的關鍵字組合。然後將它對應到操作主題，並分配給適當的代理人做執行。微軟線上說明手冊上面的逐步操作說明可以用來做為操作主題的格式。為了正確地執行操作主題內每一個指令，IDA 必須同時分析整個指令的語句，將每個關鍵的動作對應正確，如「開啟一個新視窗」，「按下按鈕」，或是執行某個應用程式等等。

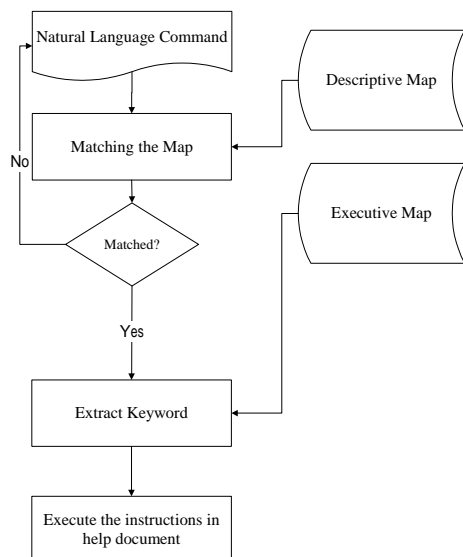
以螢幕解析度做為範例，使用者最有可能下達的指令是「我想要改變桌面大小」或是「我想要調整解析度」，而 IDA 會分析整個語句，取出其中的關鍵字：「**改變**」，「**調整**」，「**桌面**」，「**解析度**」。而根據這些詞句的組合來對應線上說明手冊，一旦成功地對應之後，會執行第一步的「開啟 **控制台** 中的 **顯示**」，因此會先開啟 **控制台**，接著再開啟 **控制台** 中的 **顯示**。由於 IDA 需要知道使用者真正需要調整的解析度的大小，而這個是屬於互動性質的，因此 IDA 就會歸還控制權給使用者，由使用者做最後調整，完成所需要的設定。

2.4 IDA 系統架構

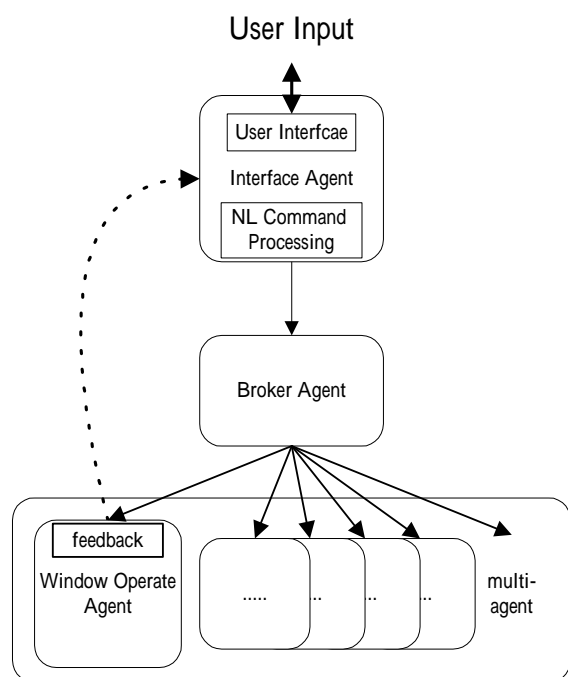
IDA 分為四個部份，第一個部份是界面代理人，它接受使用者的輸入需求；第二部份是分析一般性的操作敘述，並整理轉換為知識庫 (Descriptive Map)，做為各代理人的知識來源；第三部份則是 IDA 依據使用者的自然語言輸入對應到操作主題，並交由不同的代理人執行指令，最後則是各代理人根據自己接受到的訊息來執行任務。

當使用者輸入自然語言指令，IDA 會嘗試選粹出有用的資料並試著與 Descriptive Map 做對應。如果對應失敗，也就是代表 IDA 無法辨別使用者的指令，則將會回傳一個錯誤訊息提醒使用者重新輸入。否則，IDA 將會依照 Descriptive Map 內的步驟將關鍵字取出，並讀取 Executive Map 來決定該逐步執行那些動作或是指令。(圖一) 接著 IDA 會依照 Executive Map 而將需要進

行的指令分配給相對應的代理人執行，為了能讓代理人之間的資訊交流，我們使用了 KQML-like 的語言，以將訊息以 XML 包裝起來，由分配代理人來根據不同的任務交付不同的代理人。（圖二）



圖一. 使用者的自然語言指令對應到 Descriptive Map，IDA 將會取出其中的關鍵字內容並依 Executive Map 來執行指令。



圖二. Structure of IDA

整個 IDA 的架構重心在於正確的分析使用者的自然語言指令並能幫助使用者依據不同的任務交由代理人執行指令。

3. 實作 IDA 系統

3.1 界面代理人 (Interface Agent)

我們設計了一個對話介面讓使用者可以輸入自然語言的指令，同時為了能做教學的作用，IDA 完成一個操作主題的方法，是模擬真人操作電腦的情形來下達指令的。讓使用者看到實際執行這些指令時會看到的畫面，而不是直接完成這些工作。（圖三）



圖三. IDA

3.2 Descriptive Map

Descriptive Map 的功能在於將使用者的需求映射到已有說明文件的操作主題上。由於各作業系統的線上說明文件格式都不一，因此我們設計了 Descriptive Map 來存放格式化後的說明文件，對於一些自定的操作主題，也能有一定的格式。如此一來，IDA 針對不同的作業系統環境，都能夠轉成可攜式 (portable) 的效果。在這種情形下，IDA 只需要更改一些程式設定，而不必再重新更新 Descriptive Map 及 Executive Map。

首先，我們以微軟視窗作業系統的線上說明手冊做為主要範例，轉換進 Descriptive Map，使得 IDA 能夠理解線上說明手冊的內容。以線上說明手冊內的“**建立虛擬私人網路(VPN)連線**”做為範例，IDA 會讀取右欄的每個項目：“1. 開啟網路和撥號連線”，“2. 連按兩下 **建立新連線**，再按 **下一步**”到最後一步驟“8. 請輸入此連線的名稱，再按一下 **完成**”，並以此來建立 Descriptive Map。因此，IDA 採取的策略是將中括號中間的關鍵字詞及藍色加底線的關鍵字詞取出。並根據原來的順序放入 Descriptive Map。當使用者輸入的自然語言能夠正確對應到 Descriptive Map，那就能夠依照順序逐步代替使用者執令或是開啟視窗。對應的方法是將一個 Windows 的設定或操作的目標視為一個 FAQ，然後依據 InfoMap[8] 的 FAQ 對應的程序，而透過 InfoMap 來當作中文自然語言的語意理解核心。

為了建構 Descriptive Map，我們需要知道所有線上說明手冊語法的格式。對於微軟視窗系統，我們注意到在說明文件裡，需要被執行的按鈕或是需要被開啟的視窗都會用中括號"[]"包起來。我們同時也注意到所有需要被執行或開啟的視窗的前後文都會有"按一下"，"然後按下"，"按下兩次"等等的關鍵字。

根據線上說明手冊的敘述，有些情況是需要使用者選擇其中一個選項的，這類敘述通常都會以"如果，若是"或是一個小黑圈"做為開頭。使用者可以選擇其中一個選項來執行。除此之外，開頭為"鍵入"則是需要使用者輸入特定的資料以完成整個設定。(表一)

1.	開啟 [控制台]中的日期/時間。
2.	在 [日期時間]索引標籤上選取想要變更的項目。 <ul style="list-style-type: none"> ○ 若要變更小時，請選取小時，再按箭號以增加或減少數值。 ○ 若要變更分鐘，請選取分鐘，再按箭號以增加或減少數值。 ○ 若要變更秒鐘，請選取秒鐘，再按箭號以增加或減少數值。 ○ 若要變更 AM/PM 指示器，請選取它後再按箭號。
3.	在 [時區]索引標籤，在地圖上方的方塊，按一下目前的時區。

表一. Windows 線上說明手冊的 變更電腦時區

我們以實例來說明 Descriptive Map 的架構(圖4)。Descriptive Map 是一個樹狀的 Map，每個節點都有它的意義在，如"分類"是代表該節點之下的所有節點的分類。"屬性"則代表了該節點本身的屬性。Descriptive Map 本身包含了多個線上說明手冊的分類，如微軟作業系統的 Windows 2000，Windows 98，及 Windows ME 等等，或是其他特殊的操作主題事件。舉例來說，在 Descriptive Map 之下的分類可以找到 Windows 2000。在此分類之下我們存放轉換過後的 Windows 2000 版本的線上說明手冊。

在 "Windows 2000" 這個節點下面，有個 "屬性" 的節點，表示了所有 Windows 2000 下面的一些屬性如"使用者密碼"，"數據機" "顯示器"等等。而每個屬性下面則有"事件"的節點，配合裡面的 "變更" 節點，這代表了在 Windows 2000 裡會變更的所有事件。而有的節點下還有接著"近義語"，這是因為人們對於一個制式化名稱的稱呼可能會有類似意思的詞語來替代。如 "顯示器" 和 "螢幕" 就是同義的詞，"變更" 和 "更改" 也是近

義語。



圖四. Partial view of the Descriptive Map

當使用者的指令對應到該節點之後，如圖四中的顯示器，接著會以讀取節點 "FAQ" 下面的敘述，並且根據 "ExecuteSteps" 節點的內容來一步一步執行，如 "調整螢幕" 就會先執行 "控制台" 再執行 "顯示"。而如何執行 "控制台" 及 "顯示" 則會由 Executive Map 來做決定。

3.3 Executive Map

在 Descriptive Map 裡的 ExecuteSteps 下的每一個節點依序代表了執行的所有動作，每個動作都會參考到 Executive Map。我們定義了每個動作的控制碼，控制碼是以四位數的整數來做編碼，並且是唯一的。每個執行碼代表了 IDA 所要採取的每一個動作，如 Executive Map 中 "控制台" 這個節點裡的 "ecode" 內含的四位數字就代表當 IDA 接收到這個控制碼，它將會由螢幕左下角的"開始"按鈕開始，進到"設定"裡，再打開"控制台"視窗。也就是說，實際上節點的意義不僅只是"控制台"三個字而已。



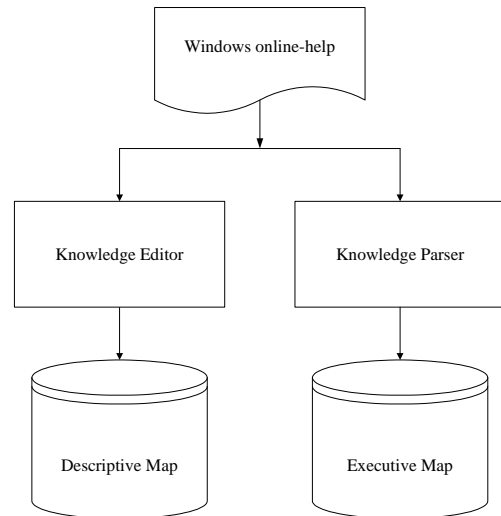
圖五. Partial view of the Executive Map

圖五是 Executive Map 的實例，與 Descriptive Map 同樣為分類過的資料，根據事件屬性來做分類，如 Windows 類別裡會對系統 (System) 的或是視窗 (Window) 的再做分類，系統內存放著整個作業系統都會呼叫到的事件，如"控制台"視窗裡的元件。而視窗裡存放的是對於單一視窗的事件控制，如"上一步"，"下一步"，"完成"等等。

3.4 建立 Map 的方法

關於建立 Descriptive Map 與 Executive Map 的方法，我們是在使用 IDA 之前預先使用一個知識的解析器 (knowledge parser) 對線上說明手冊做處理來獲得資料。我們知道線上說明手冊在提到需要使用者執行的動作，都會將該動作或指令以並"[]"包圍起來，或是以藍色字呈現，因此我們的 knowledge parser 是處理線上手冊的文字，並將所找到的指令或動作視為關鍵字。接著對於每個關鍵字依序產生四碼的數字代碼，最後再將正規的關鍵字及代碼存放到 Executive Map 裡供底層使用。由於實驗主要目的在於展現 IDA 能自動代替使用者執行一些指令，目前對於 Descriptive Map 的做法，是由人工輸入的方式，以同等於線

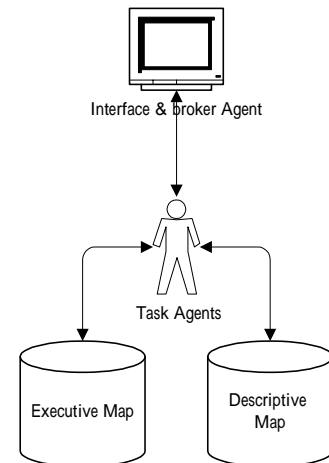
上手冊的步驟來相對應輸入。



圖六. Knowledge Parser 讀取 Windows 線上說明文件。分析之後產生 Executive Map。而 Descriptive Map 由人工建立。

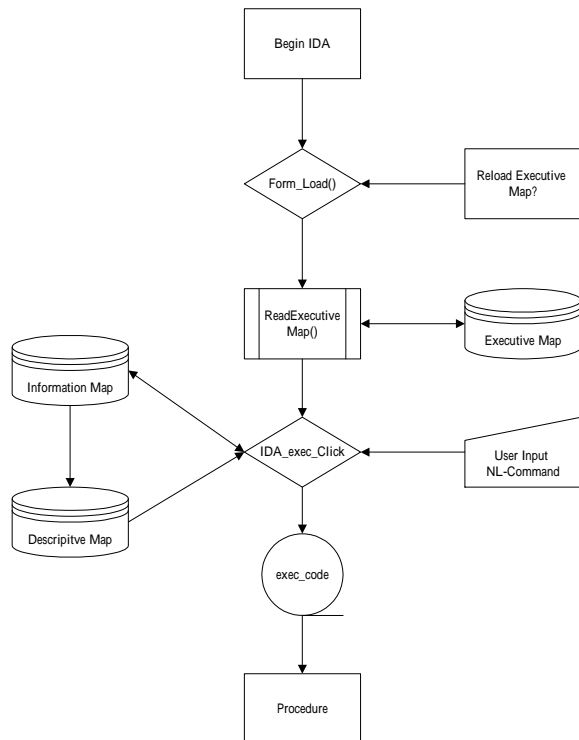
3.5 程式流程

整個程式是由一個界面代理人配合之下的多代理人組成。啟動 IDA 後會讀取 Executive Map 的資料，並將取得的資料儲存在記憶體。而任務執行代理人會依照 Map 的資料及界面代理人傳入的訊息執行必要的指令。



圖七.代理人讀取 Map 獲得知識並執行指令

3.5.1 Read Maps & Parse



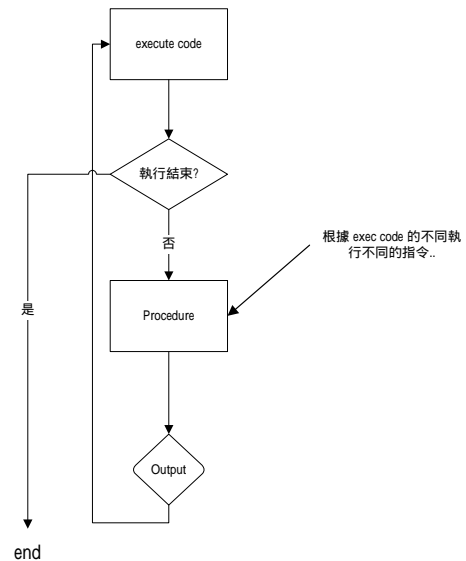
圖八. 程式流程(1) Read Map and Parse

IDA 一執行就會先行將 Executive Map 讀入 E_Nodes, 而 E_Nodes 記錄著所有的事件發生的代碼(如:“完成”,“Enter”,“控制台”)。IDA 執行的全程並不會有對於 Executive Map 有任何更改的地方,因此我們設定為在程式執行初期便先行載入。如此便可以加速 IDA 執行的速度,而不必在執行每一步指令時還需要對整個 Executive Map 做存取。

而在將 Executive Map 讀入記憶體之後,IDA 便開始接受使用者輸入的自然語言指令。當 IDA 接收了自然語言指令,將會呼叫 InfoMap 元件來處理自然語言指令。在得到 InfoMap 分析自然語言指令的資訊之後,IDA 接著試圖對應到 Descriptive Map,如果沒有對應到則會傳回空的結果值,會在 IDA 上面的指令執行視窗顯示“找不到相對應的指令”。

3.5.2 Execute Steps

當 Descriptive Map 被對應到時,會在被對應到的節點下面找出需要 IDA 執行的項目,再由 Executive Map (先前讀進 E_Nodes 中)轉換出所要發生的事件代碼,接著再逐一透過多代理人來執行。代理人的運作方式是負責對所接收到的代碼作相對應的處理動作,如“開啟視窗”,或是“按下 Enter”等等動作。



圖九. 程式流程(2) Execute Steps

4. 結論及未來工作

4.1 結論

目前完成的 IDA 系統配合了 InfoMap 系統,有能力接受電腦使用者的自然語言指令來操作電腦。設定電腦的知識是取自於線上說明手冊,而 IDA 可以把電腦使用者的自然語言指令對應到線上說明手冊的命令敘述並執行相關指令。整個執行流程重心在於讓自然語言指令可以對應到 Descriptive Map,再根據 Descriptive Map 來決定執行那些指令或是動作,而決定指令的執行動作則由 Executive Map 來定義。在這個架構下,我們只需要將知識寫入 Descriptive Map,而使用者只需要擁有基本的使用電腦知識就可以具有操作電腦細部設定的能力。把使用者對於電腦操作使用的困難之處轉換到建立 IDA 內部 Descriptive Map 的知識量,以及多代理人對於 Executive Map 的處理能力。

4.2 未來工作

由於現在 IDA 尚只能處理單一流程的自然語言指令,在遇到一些需要使用者再輸入資料的情形,就只能停在該步驟。希望可以利用對話式的交談,在某些需要使用者輸入資料的時候,跳出對話框或是等待使用者輸入於現有的視窗,並繼續進行下一步的指令。

同時我們也希望 IDA 能具備個人電腦上知識管理的功能,例如一個小的搜尋引擎來找尋電腦內部的所有文件資料,或是能接受使用者的查詢關鍵字,每日能固定上網搜尋相關的資料並整理。而這些資料的存放可以在新建立的 Map 裡。語音輸入的技術在目前已經相當成熟了,未來也

可以結合語音輸入，使得 IDA 不需要再利用鍵盤，更可以增加 IDA 的適用性。

除此之外，我們可以記錄下使用者透過 IDA 所執行的每個指令，進而可以在使用者使用 IDA 時，對下一步動作給予建議。也可以分析出使用者的偏好，在使用 IDA 時就預先替使用者找出所需的相關資料。

參考文獻

- [1] Suhayya Abu-Hakima, Connie McFarland and John F. Meech; An Agent-Based System for Email Highlighting, Proceedings of the fifth international conference on Autonomous agents, 2001, Pages 224 – 225.
- [2] Joyce Chai, Veronika Horvath, Nicolas Nicolov, Margo Stys-Budzikowska, Naturla Language Sales Assistang – A Web-based Dialog System for Online Sales, Proceedings of the Thirteenth Innovative Applications of Artificial Intelligence Conference, August, 2001
- [3] Tim Finin, Richard Fritzon, Don McKay and Robin McEntire, KQML as an Agent Communication Language, Proceedings of the Third International Conference on Information and Knowledge Management, ACM Press, November 1994.
- [4] Godwin M. Gwei & Eric Foxley, Towards a Consultative On-Line Help System, International Journal of Man-Machine Studies 1990 v.32 n.4 p.363-383
- [5] I. Greif, Desktop Agents in Group-Enabled Products, Communications of the ACM, 37(7), pp. 100-105, July 1994.
- [6] Wen-Lian Hsu and Yi-Shiou Chen, On Phoneme-to-Character Conversion Systems in Chinese Processing, Journal of Chinese Institute of Engineers 5, (1999), 573-579.
- [7] Wen-Lian Hsu, Yi-Shiou Chen and Yuan-Kai Wang (1999), Natural language agents - An agent society on the Internet, in Proceedings of PRIMA 99.
- [8] Wen-Lian Hsu, Shih-Hung Wu, and Yi-Shiou Chen, Event Identification Based On The Information Map - INFOMAP, to appear in symposium NLPKE of the IEEE Systems, Man, and Cybernetics Conference, 2001.
- [9] H. Lieberman, Autonomous Interface Agents, ACM Conference on Human-Computer Interface [CHI-97], Atlanta, March 1997.
- [10] P. Maes, Agents that Reduce Work and Information Overload, Communications of the ACM, 37(7), July 1994.
- [11] Microsoft Agent Home, <http://msdn.microsoft.com/msagent/>
- [12] Jeanne Murray, David Schell and Cari Willis;

User Centered Design in Action: Developing an Intelligent Agent Application, Proceedings of the 15th annual international conference on Computer documentation, 1997, Pages 181 - 188

- [13] Shih-Hung Wu, Min-Yuh Day, Wen-Lian Hsu, FAQ-Centered Organizational Memory, Proceedings of IJCAI'2001 workshop on Knowledge Management and Organizational Memory, 2001.