# Integrated Application Agent-Based Services

**Susmini Indriani Lestariningati**
Computer Engineering, Faculty of Engineering and Computer Science, Indonesia Computer University

**Keywords:**

Software Agent
Intelligent Agent
User Profile
Application Service Provider
Cloud Computing

## ABSTRACT

This paper is aim to give illustration how the software or application services in the cloud go further and propose the original architecture frame work of this application services interact and will be integrated. It is also elaborate the integrated deployment of intelligent software agents in many areas, from system and application interfaces using agent as personal assistant to access of various application, communication, information, consultation, and many other services with user profiling enhancement.

The original model design proposed is based on multi-agent architecture, which consists of 3 categories of agent: personal agent, service agent, and information agent, plus administration agent. Inter-agent cooperation and communication is supported using existing interoperability mechanism. Implementation of agent follows object-oriented paradigm using components as agent's building block. This is done so to enable reuse and cloning of existing agents.

Distribution of agents follow cloud-based model, where an agent is accessed per need basis. An agent matchmaker, who provides list of agents along with their service offerings, is used to assist in finding the required agent. The sample scenarios of implementation also present to illustrate the further developments.

***Corresponding Author:***

Susmini Indriani Lestariningati,
Computer Engineering, Faculty of Engineering and Computer Science
Indonesia Computer University (UNIKOM)
Jalan Dipati Ukur No.112-114, Bandung, Indonesia.
Email: lestariningati@yahoo.com

## 1. INTRODUCTION

The rapid development of information and communication technology (ICT) offers opportunities with an ever increasing and more sophisticated application types, information coverage, and devices variety being used to access the technology. Deployment of the technology in its various sorts is expected to help people carrying out their activities. Yet, its accessibility is not as pervasive as it promises due to several inherent problems.

Different level of competence among users using the technology complicates its accesses. Often, a computer system or application is designed assuming certain level of user proficiency and discourages accesses by users below that level. This situation could lead to an even broader technology gap among users. Many approaches were introduced in order to ease access of such systems; among them are GUI (*Graphical User Interface*) and voice-based commands. Such interfaces turn out to be awfully complex, yet often ignore the essential problem of providing interfaces that are not only user-friendly but also user-assisting for supporting interaction with the systems.

The enormously high-cost of licensing IT products, particularly application software, inhibits widespread access of the technology as well. The price paid by users accessing a software product is equal regardless of their usage frequency. Besides, users usually not utilize all features offered by software optimally, worse yet; most users only use a few simple features. Thus, the utilization of software is often not comparable to the cost paid for its purchase.

This paper discusses a novel approach to eliminate the problems mentioned above using combination of software agent technology and online application service provider (ASP) concepts. Software agent is a special kind of software acting on behalf of users to assist doing their routine tasks. Online

Application Service Provider is, in fact, we propose a model of software distribution in the cloud as in Figure 1, which deviates from the conventional way in which software is installed only as needed.

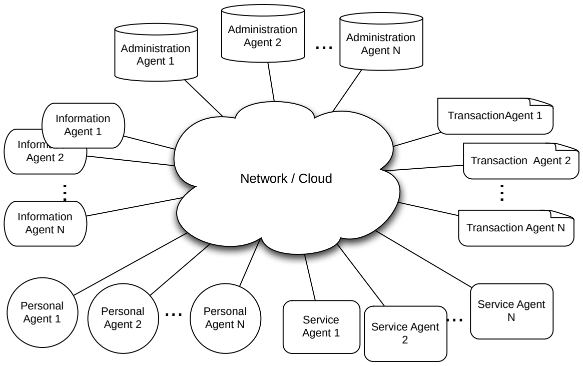

Figure 1. Software Agents in Cloud

Thus, software is charged based on usage frequency (pay-per-use), much like charging of telephone calls. In relation with agent, the main benefit offered by online application services provider is its ability to provide wide range of services, in form of application programs, consultation, or information, customized according to specific user needs and characteristics (user profile).

In the parts below, we will discuss several fundamental characteristics of agent required to develop intelligent agents, which could perform specific tasks according to their functionalities. Next, we will highlights on components that build up the system architecture along with short description on agent learning mechanisms.

## 2. AGENT COMPONENTS

In this section we describe in details how software agents in cloud works and what it can achieve, by describing the functionality of the main agent components and following an accompanying scenario.

## 2.1 Software Agent

### 2.1.1 Agent Characteristics

There are several fundamental characteristics need to be acquired by intelligent agents:

- Taskable, in which agents could perform certain tasks requested by human or other agents;
- Network-centric, in which agent is distributed and mobile;
- Semi-autonomous, with self-control ability so as not to insist on human intervention;
- Persistence, with long-term operations and data storage;
- Trustworthy, where agent could perform user tasks reliably and secretly;
- Anticipative, with the ability to anticipate user needs based on current role and situation as well as the ability to learn;
- Active, the agent has the ability to initiate problem solving tasks;
- Collaborative, with humans or other agents;
- Ability to work with diversity, either diverse user, information sources, or work environments;
- Adaptive, to user needs and work environment.

These are common characteristics need to be considered for agent implementation.

### 2.1.2 Agent Learning

To facilitate deployment of an agent as intelligent assistant, which could help user based on his needs and characteristics, we need to supply the agent with necessary knowledge on its user and application domain. There are 3 approaches to enable agent acquires such knowledge:

- User actively involves in providing rules on certain tasks and the agent, subsequently, performs such tasks User actively involves in providing rules on certain tasks and the agent, subsequently, performs such tasks based on the given rules;
- Uses knowledge engineering technique to identify user and application domain. This technique is not customized for every user, but for each category of users according to the previously acquired knowledge;
- Build user profiles describing his preference to certain application and services.

In our model, knowledge acquisition for the agents will incorporates all positive aspects from the above approaches, with main focus on the development of user profiles, in order to acquire more comprehensive domain knowledge. Early phase of knowledge acquisition will concentrate on the study of application domain and user categories using knowledge engineering techniques to attain basic knowledge used as the baseline for building profile for each specific user. User then provides, using rules, his preferences for certain applications, services, and information to result in more specific profile. Next phase will be the refinement and enhancement of existing profile in form of agent learning, through experience, observation, and user interaction, in order to better comprehend user requirements and adapt to user needs and preference changes dynamically. In addition, agents also store user profile history to be retrieved as needed. We propose in Figure 2 describes the acknowledge acquisition process in more detail.

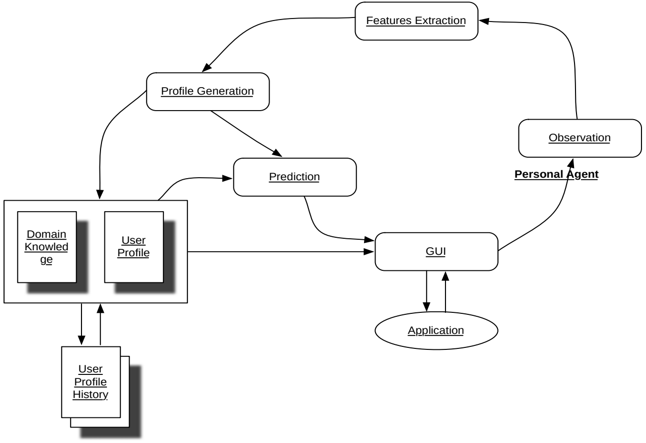

Figure 2. Agent Learning Process

There are several mechanisms could be used to produce specific profile, among them are genetic algorithms, symbolic rule induction algorithms like C4.5 and CN2, and neural network. These mechanisms will be chosen, based on their effectiveness and learning period required, to produce the best performance.

### 2.2 System Architecture

System architecture is structured modularly using multi-agent system approach where system functionalities are decomposed into several agent component categories, each of which performs specific function. In each category, agent could be further decomposed based on domain and problem type. Multi-agent system posseses several advantages as described below: Focus an agent for one specific task to enable it to function effectively and not being overloaded by multiple simultaneous tasks;

- Lean agent structure, especially required for mobile agents to easily move;
- Ease for development of new agents or cloning of existing agents through reuse of agent components and their knowledge;
- Reduce heavy processing load in one agent to avoid bottleneck and single point of failure.

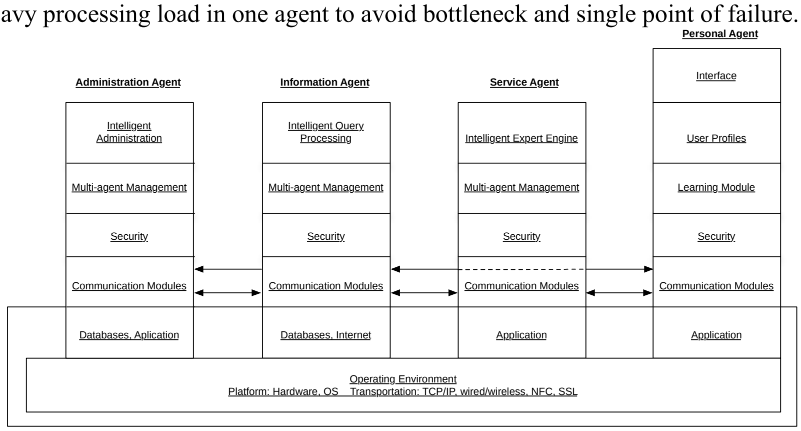

Figure 3. Proposed System Architecture

Multi-agent system, in this case, implies that an agent not only interact with its human users but also with other agents to request their services or information. This asks for good interoperability mechanisms in forms of communication, interaction, and collaboration among agents, either synchronously or asynchronously. These mechanisms should be done transparently to result in a seamless system.

### 2.2.1. Agent Components

As shown in Figure 3, the system will consist of 4 categories of agent components: Personal Agent which directly interacts with user to interpret his needs, Service Agent to provide specific functionalities and services, Information Agent which functions as information provider, and Administration Agent which function as administrations of software agents.

Personal Agent plays role as a personal assistant, which directly interacts with user to help him doing his tasks. The main functions of personal agent are: (1) collect information from user to start its work, (2) interact with user during the task execution, and (3) display and report task result to user in an easily understandable format. Personal agent interacts with user based on its acquired user profile model. During the interaction process, personal agent also performs continuous learning to enhance its knowledge model or update it, if necessary.

Service Agent provides specific services to users or other agents. The services offered could be in form of application, communication, decision support, expert advice, and many others. Since the types of potential service that could be offered are exceedingly broad and the service processing is quite complicated, service agent modeling turns out to be complex and thus generating common architecture for all services is not straightforward. In general, service agent functions to: (1) receive task specification from personal agent, (2) identify task objectives based on the specification and derive plan to execute the task, (3) decompose task into several subtasks and distribute them to other service agents, (4) if necessary, coordinate information agent to search for information, (5) compile the results obtained from task decomposition into an integrated services, and (6) deliver the task result to the user through personal agent.

Information Agent is a category of agents that functions, actively or passively, as information provider for other agents (personal agents and/or service agents). Information agent could indeed be derived from service agent, yet, in this case, the agent focuses on using specific problem solving techniques for information searching. This is because there exists so large and wide variety of information in the internet/cloud world that it requires special handling to retrieve the information timely and accurately. In essence, the information is provided in one of the following ways: (1) incidental query for requesting specific information, (2) periodic query that runs repeatedly in certain period of time, e.g., extracting daily news from an internet/cloud site, or (3) monitor an information source and notify user if any change occurs, e.g., information on changing of certain currency values.

Administration Agent is a Software agent's provider that develops and delivers the other agents to be implemented as Personal Agents, Service Agents, and Information Agents. The Administration Agent also handling the registration, authentication, and problem handling of Agents performance.

### 2.2.2. Interoperability

In performing its task, an agent interacting within multi-agent system usually needs to communicate, coordinate, and collaborate with other agents, either synchronously or asynchronously. These agents could be in the same location, while, in many cases, the agents are distributed in different and remote locations that they require specific mechanisms for their interoperability.

In relation to system interoperability, there are several issues need to be considered:

- Task decomposition, to apportion task into smaller subtasks and assign them to other agents;
- Coordination, using an agent appointed as the coordinator or distributedly coordinates the tasks without a single coordinator;
- Inter-agent communication, using such mechanism as message passing;
- Sharability, to share services among several users/agents for efficient use of resources;
- Complexity hiding, which conceals the complexities of delegating tasks to several agents and distributing agents in various location;
- Flexibility, to allow an agent to cooperate with other agents alternately without requiring any substantial change;
- Robustness, to allow the system to continue working even in the failure of several agents; and
- Security, to protect against illegal data accesses.

Based on the above criteria, agents are expected to be able to perform tasks autonomously based on need and to minimize user intervention. Inter-agent communication could be accomplished through such agent communication languages as KQML (Knowledge Query Management Language). The model of agent interoperability is shown as in Figure 4.

The interoperability between Administration Agent will depends of each Agents Provider will open up to each other. However the interoperability could be difficult when there is no common architecture standard when agents is defines and interact. This paper is proposed on of option the architecture of agents that can be a based for multi-agents providers interoperability.

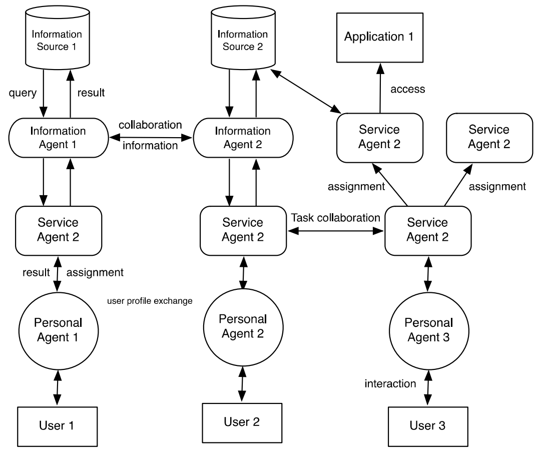

Figure 4. Agent Interoperability Model

### 2.2.3. Operation Environment

On the bottom of the system architecture lays the system basic environment, consisting of hardware, operating system, and network that provides mechanisms for inter-agent communication, data delivery, and agent movement. There are large variations of supporting hardware, from conventional PC, notebook, Tablet, mobile phone, to various embedded devices. To allow flexibility, an agent should be able to run in various operating platforms such as Windows, Unix and their variants. For communication, the agent could use such transport layer mechanisms as TCP/IP, wireless, SSL, NFC, and serial connection.

### 2.2.4. Integration with Applications

On top of the basic environment, in the system architecture, are various application programs interacting directly with agents. Here, the agents carry out their main tasks of assisting user use various application programs, providing numerous services, and accessing variety of information available, all of which require interaction with application programs. The application programs employed are those already available in the market, not specially-designed applications built from scratch. The reasons are:

- To take advantage of facilities and capabilities of various existing application software;
- To facilitate user preferences for certain application products;
- To focus more on agents building, not on application development.

Interaction between agents and applications has unique characteristics for each category of agent component. For personal agent, the focus is on how the agent could comprehend user specific needs based on existing profile to assist user use applications. An example is a word processing application in which the agent will observe and infer about features normally utilized by the user for various purposes so as to allow the agent to provide suitable template for user needs. Another example is a mail reader application agent which filters incoming mail to discard any junk mail, categorizes and saves them into appropriate folders, and notifies user of any important mail entering the mailbox. For service agent, its interaction with application program is very specific to service provided by the agent. For example, expert assistant agent in marketing could provide expert advice on marketing strategy specific to user problems based on existing expert system engine. For information agent, the interaction is with information provider application such as database and website.

One mechanism for interaction between agent and application program is through access of application internal data structures. Although internal data structures of existing applications could hardly be accessed directly, yet applications usually provide access mechanisms via API (Application Programming Interface) or scripting language.

### 2.3 Agent Implementation

Agent will be developed using object-oriented paradigm, where the composite elements consist of inter-related components. This is intended to support rapid development of agent using reuse principle of existing components. This is likely since agents of same type usually have similar functionalities that it is not necessary to built each new agent from scratch. Similarly, agents of different categories could be built from several equivalent components as shown in Figure 3. There are 2 approaches for agent implementation:

- *Top-down*, where a framework containing generic design of all agents is developed prior to agent implementation by agent providers. Based on this framework, basic components of agents are developed which then forms the basis for agent development (could be used ‘as is’ or after modification);

- *Bottom-up,* where implementation starts from development of specific agents that later, based on common characteristics of their designs, are used as models for developing framework and repositories of agent components. The framework could be used as reference for designing subsequent agents and usage of existing components.

## 3. SCENARIOS

After discussing agent architecture and implementation, in order to provide more descriptive explanation on this topic, this part will present two simple agent scenarios. The first scenario describes personal agent implementation for assisting user doing his marketing tasks. The second one gives an example on the utilization of a service agent for providing integrated communication services to its user.

### 3.1. Scenario 1 : Personal Agent

Personal Agent (PA) is the main agent, which interacts directly with its human user. To assist its user performing his tasks, PA need to have user profile acquired through learning mechanisms as explained in section 2.2. In this case, the user is a salesman who performs various marketing-related tasks. Based on this profile, PA will assist its user based on his needs and preferences.

The scenario is as follows:

- The PA will be activated as user boots his End User Terminals;
- PA will examine the system's last status, any incoming mail or message or any other communications, and reminder to be displayed;
- Based on user profile, the agent will display interface appropriate for user characteristics. On this interface, the agent will display information on incoming mail/message and reminder for its user;
- Based on the acquired user's periodic information needs, PA will access associated service agents (SAs) to assist it performing the tasks. For example: if every Monday the user needs to generate a report on last week's recruitment of new customers, the agent will access word processing SA and customer's database SA to assist the user.
- The retrieved SAs will communicate with the PA asking for its user profile. Based on the profile, SAs will download an application with template and interface suited to user characteristics and needs so as to ease the user in generating the report;
- Once a time, the user might require non-routine services such as consultation on marketing strategy. PA will look for an SA providing such services and inform the SA on its user profile. Based on this profile, the SA will provide marketing consultation to the user;
- PA will also display information on potential customers obtained from information agent (IA). Accordingly, PA will help the user contacting the to-be customers via email.

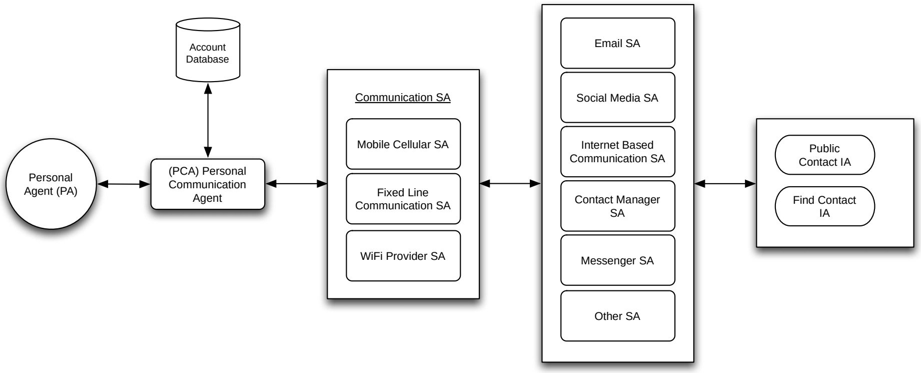

Figure 5. Personal Communication Agent Implementation Scenario

### 3.2. Scenario 2: Personal Communication Agent

Personal Communication Agent (PCA) is a service agent whose function to provide integrated communication services to its user. Any connection setup, either voice communication via fixed/mobile telephone or data communication via SMS, mail, messages, or fax, has to go through PCA. PCA also stored and coordinate all contact lists of several devices and several accounts. In this simple example, the user is a businessman who frequently moves from one location to another, yet requires non-interrupted communication connection with all his partners. The scenario is as follows:

- Based on PCA observation, the pattern of user movement and the available communication facilities on each visited location is acquired. The user's current position could also be identified if his cellular phone/mobile tablet that is presently active;

- PCA will manage the whole communication channels that available of person has and could connect user to by the most economical and reliable such via a local number of his present location, if possible;
- If the user receives a fax but his present location does not have any fax machine, the user could read the fax content via email, or any picture based message services available, which is enabled through data conversion service provided by SA; Similarly, any incoming e-mail for the user could be converted into voice mail accessible by user via phone;
- If the user is on a meeting, PCA will filter any incoming call and will only deliver an urgent one to the user;
- If the user requires information on phone number of a business partner, the PCA could access IA to find information.

## 4. CONCLUSION

This paper proposes the original model architecture on the potential utilization of agent technology for advance development of IT. There are many issues which could be further developed using this agent technology, from the development of more intelligent user and application interfaces using agent as personal assistant to the access of various such services as communication, information, and consultation suited to user characteristics and needs (user profile). To be able to perform its task intelligently, an agent must be equipped with learning capabilities on user and problem domain.

Software Agents are designed using multi-agent system, due to complexity of the problem they might handle, which that is impractical to solve it by a single agent. The likely high demand for agents is anticipated using model that enables rapid development of agent. This could be done through reuse mechanism and cloning of existing agents. For this purpose, agents will be developed based on object-oriented paradigm using components as their building blocks. The interoperability between Agents Provider will depends on the providers, if necessary the standard of architecture of how agents interact can be defined.

Agents and their applications/services are distributed using Cloud concept where they will only be accessed as needed. This allows for saving of storage needed to store such agents and their applications/services as well as lowering software cost compare to conventional distribution model. To assist in finding suitable agents, an agent matchmaker, who provides list of available agents along with their service offerings, is deployed. This paper also provides two sample scenarios of agent deployment for the purpose of providing more descriptive explanation on this topic.

## REFERENCES

[1] K. Mong Sim, "Agent-Based Cloud Computing," Journal of IEEE Transactions on Services Computing, vol. 5, no. 4, pp. 564-577, 2012.

[2] N. Griffiths and K-M Chao, "Agent-Based Service-Oriented Computing (Advanced Information and Knowledge Processing)", Springer, 2010.

[3] T. Konnerth, “An Agent-Based Approach to Service-Oriented Architectures”, Desertation of Elektrotechnik und Informatik der Technischen Universit ̈at Berlin zur Erlangung des akademischen Grades, 2012.

[4] P. Edwards, “Intelligent Agents = Learning Agents”, J L Nealon & N S Taylor (Eds), Proceedings of the UK Intelligent Agents Workshop, SGES Publications, Oxford, 157-161, 1997.

[5] P. Edwards, D. Bayer, Claire L. Green and Terry R. Payne, “Experience with Learning Agents Which Manage Internet-Based Information”, M A Hearst & H Hirsh (Eds), AAAI 1996 Stanford Spring Symposium on Machine Learning in Information Access, SS-96-05, AAAI, 31-40, 1996.

[6] H. Lieberman, “Integrating User Interface Agents with Conventional Applications”, Knowledge-Based Systems Journal, Elsevier, Vol.11, No.1, pp. 15-24, September 1998.

[7] S. Schiaffino and A. Amandi, “Intelligent User Profiling”, Artificial Intelligence An International Perspective Lecture Notes in Computer Science, Volume 5640, pp 193-216, 2009.

## BIBLIOGRAPHY OF AUTHOR

| | |
|---|---|
| | Susmini Indriani is a lecturer at Computer Engineering in Indonesia Computer University (UNIKOM) Bandung from 2007. In 2002, she finished the Undergraduated Degree in Electrical Engineering from Atma Jaya University in Jakarta. In the year of 2006, she finished her Master Degree from Institute Technology Bandung (ITB) at School of Electrical Engineering and Informatics majoring in Telecommunication Information System. |