

## Research Article

# Ontology-Based, Process-Oriented, and Society-Independent Agent System for Cloud Computing

Mahmood Fathalipour,<sup>1</sup> Ali Selamat,<sup>1</sup> and Jason J. Jung<sup>2</sup>

<sup>1</sup> *Software Engineering Research Group (SERG), Knowledge Economy Research Alliance and Faculty of Computer Science & Information Systems, Universiti Teknologi Malaysia (UTM), 81310 Johor Bahru, Johor, Malaysia*

<sup>2</sup> *Department of Computer Engineering, Yeungnam University, Gyeongsan 712-749, Republic of Korea*

Correspondence should be addressed to Ali Selamat; [aselamat@utm.my](mailto:aselamat@utm.my)

Received 13 December 2013; Accepted 24 December 2013; Published 12 March 2014

Academic Editor: Yue Shan Chang

Copyright © 2014 Mahmood Fathalipour et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IT systems are used to help organizations to manage and automate their processes. However, most of today's systems are not reusable because of mixing the society's knowledge with the process's knowledge. Since the societies' knowledge is different to each other, the applications are not reusable. Hence, this paper address the dependency of the applications on the societies by separately defining process's ontology, the agent's knowledge, the society's ontology, and the society's knowledge. This research introduces an ontology-based, process-oriented, and society-independent agent system which allows all the organizations to utilize it by defining and importing their societies' ontology and some process patterns, which can be instantiated from the process's ontology, into the system. The proposed system can be applied to cloud computing platform. The proposed system has been evaluated from two perspectives, the quality by using the cohesion and the coupling measures for measuring the degree to which the system focuses on solving a particular problem and the applicability by evaluating the manageability and the automatability of the seven processes from three different societies. We believe that our proposed system will improve when we apply some methods to find the best process patterns and perform parallel processes.

## 1. Introduction

Today's marketing is very intensive; therefore, it is needed for organizations to try to reduce their manufacturing cost and increase their benefits from the market. As [1] mentioned, organizations, in order to make profits, need to sell their products. They advertise their product in order to maximize their profits and try to keep their costs low using traditional ways. However, these approaches have limitations; hence, organizations need to find other ways to achieve their goals. Management and automation of the processes are two key success factors which can help organizations considerably reduce their costs and get benefit from the market. As Redwood, the leader in enterprise process automation and SAP's partner company, mentions, agility, insight, better controlling, and speed to the customers' expectations are advantages of the process automation [2, 3]. Since IT showed

itself as a suitable means for managing processes, many agents and systems were introduced to help organizations to automate and manage their processes. Agents are able to totally automate business processes [4, 5]. However, these systems are specifically implemented for managing processes of one department or division of organizations. Thus, one organization in order to manage their processes may implement agents ontology in many systems. This isolation of systems brings many problems for organizations such as the cost that organizations need to pay to plan, design, implement, and test the systems. The most important problem is that, if the organizations decide to change their processes, the systems are completely useless, and organizations need again to do all the software development life cycle's activities to produce new system which is matched with their processes. Therefore, the problem that organizations face today is the dependency of the system on the societies which followed

by two subproblems. First, they need one system for each functional area, and, second, since the processes are cross-functional and contain functional area, by changing one process, all the systems need to be changed.

The development of ontology facilitates the interoperation of information among processes [6, 7]. However, merely using ontology cannot solve the dependency problem. All of today's ontology-based systems use ontology instead of the database or as a complement of the database. Since the ontology of each society is different from others, one system is not able to communicate with more than one society.

This study seeks to introduce society-independent system which is able to be adjusted according to the organizations' changes. Moreover, since an introduced system is independent of society, it can be utilized in all organizations for managing and automating their processes. In addition, since the system uses ontology, it is understandable by machines which have automation of the processes as a result. In fact, our method tries to separate the knowledge of the process from the knowledge of the organization's environment by introducing separate ontologies, one for each process and one for each society. The rest of the paper is organized as follows. In Section 2, we discuss some ontology and process-oriented studies. Section 3 explains our proposed model. By using two metrics, some ontology of different societies, and some process patterns, we evaluate the quality and applicability of the system in Section 4. We will discuss our results in Section 5. Finally we conclude the paper in Section 6.

## 2. Related Work

As mentioned in the previous section the aim of this study is to propose a system which can manage and automate processes and is also independent of society. Ontology is a tool which by managing knowledge helps organizations to manage their processes. Ontology is the specification from the conceptualization [8]. Knowledge is a set of entities and the relation among them [6, 9]. Therefore, if we consider knowledge of a society as a set of concepts which are implicitly inside the minds of some people, ontology tries to explicit that knowledge and shows them as entities inside knowledge base (KB). In recent years ontology has widely been applied in almost all societies in order to manage the processes. Ontology also is used for automating processes. The machine which is able to understand knowledge can be obtained through ontology [10]. Once a machine can understand a process, it can therefore perform it; so the process will be automated. The ability to automate a process involves the need to represent it in such a way that is understandable to a machine [6, 11]. Unfortunately we could not find any study related to the independency of the system of the society. However by finding process-oriented studies, we found some clue to achieve our purpose. Hence, next subsections survey papers that tried to manage and automate processes and also papers which used a process-oriented approach to distribute knowledge.

*2.1. Process Management Using Ontology.* Ontology-based e-learning system which is introduced by [12] is a system to represent knowledge of e-learning society. They introduced a system which consists of five ontologies, namely, domain, task, learner, teaching strategy, and inference ontologies, to perform effective personalization of e-learning. Each of the introduced ontologies contains many concepts and the relation among them. The knowledge of universities and their departments and colleges in different states of India is explicitly presented by [13, 14] using ontology. An ontology-based system for finding implicit knowledge from the explicit data of yeast by using the reasoning engine is introduced by [15]. An ontological knowledge base that covers healthcare domains is introduced by [10] which contains almost all parts of the hospital, administrative, patient records, drugs, and hospital assets.

*2.2. Process Automation Using Ontology.* A semantic web service is a concept which is introduced by [16, 17] in order to facilitate supply chain coordination and automate the interactive processes. They proposed one ontology for the web service and called it OWL-S. According to this paper, the web service ontology is a set of service profile, service model, and service grounding. The service model is of type process which can be a composite process or an atomic process. However, this system is not able to communicate with humans. Besides, there is not any definition for process which makes its proposed system be dependent on society. A process ontology for e-business is introduced by [6]. According to Garcia, the process set consists of context, initial context, situation, action, fact, verification, and decision  $P = \{C, c', S, A, F, V, D\}$ . They used decision concept in the process knowledge in order for its agent to be able to communicate with the end user. Situation concept is considered to show the current activity that the agent needs to do. However, this definition of process does not correspond to the process definition from [18] which defines the process set as a set of objective, input, activity, and output which has an owner and a recipient  $P = \{\text{Objective, Owner, Initial information, Activity, Result}\}$ . Besides, they do not consider situations in which initial information (fact) does not exist. Also, the decision of the user cannot be considered as process concept because it is equal to the intention of the user for starting the process and belongs to the user. This system also depends on the society, namely, the sales supporting part of a telecommunication company.

*2.3. Process-Oriented KM.* A process-centered knowledge management model which is introduced by [19] retrieves information by using the process's name or id. They compare the retrieval of information by using their proposed system with information retrieval by using the keyword and show how by using ontology we are able to have multilayered information instead of one-layered by using the keyword. It can be said that they used the process as the subject of the RDF triple graph which has participants, product, and resource as its objects. However, the aim of this system is the only information retrieval and process does not have

TABLE 1: Summary of paper review.

Primary author (year)	Manage process	Automate process	Society-independent	Description
Vesin et al., (2012) [12]	✓	X	X	Use ontology to explicit knowledge
Rabiyathul (2012) [26]	✓	X	X	Use ontology to explicit knowledge
Villanueva-Rosales and Dumontier (2008) [15]	✓	X	X	Use ontology to explicit knowledge
Dang et al., (2008) [10]	✓	X	X	Use ontology to explicit knowledge
Liu and Guihua (2007) [16]	✓	✓	X	Web service = {service profile, service model, service grounding} Process = {atomic process, composite process}
Garcia-Crespo et al., (2011) [6]	✓	✓	X	Process = {Context, Initial context, Situation, Activity, Fact, Verification, Decision}
Han and Park (2009) [19]	X	X	X	Introduce process-oriented for information retrieval
Woitsch and Karagiannis (2005) [20]	X	X	X	Introduce process-oriented for information retrieval
This study	✓	✓	✓	Process = {Owner, Recipient, Objective, Context, Possible Fact, Fact, Activity, Dependency, Result}

any intervention at operational level. Process-oriented KM is categorized into three categories by [20, 21]: first, defining, managing, and distributing the processes; second, using The KM to make the first activities more efficient; and finally, managing the KM. This paper also does not use process at an operational level.

It can be seen in the above studies, that the most important problem is the dependency of the systems on the societies which they are deployed and implemented for. Some studies utilized ontology for managing and automating the process, and some used the process at the center point for connecting information. However, neither of them used process as an interface between the societies and the agent. A summary of studies which were reviewed in this research is shown in (Table 1).

### 3. Proposed Model

It can be inferred from the previous section that the real problem is the lack of a clear boundary between the societies' knowledge and the process's knowledge. Once the boundary of the knowledge is not defined, the knowledge of the societies will be mixed with the process's knowledge which has the independency of the agent as a result. In this study we tried to develop a system similar to a calculator, while it is not limited to the number of operators and type of data. A calculator does not know what the numbers, which are entered, represent. The only thing that the calculator does is receiving the numbers as the arguments and the operator as a function and then producing output by applying the function on the numbers. Thus, in order to have a system similar to the calculator and having an independent agent, this paper will clearly define and separate the knowledge of each society and knowledge of the process. In addition it will define the knowledge of the web page regarding the type of data that it needs to display to the user or receive from users.

The process ontology's role is connecting the agent to the societies (Figure 1). Therefore, the only thing that the agent needs to know is the knowledge of the process concepts. The knowledge of the societies, the web page, and the process will be represented by using ontology. The knowledge of the web page contains any things that it needs to know, for example, the type of the contents which determines how the web page must display them.

The web page obtains the information that it must display from the agent and find the type of them from the web page ontology.

#### 3.1. Ontology Definition

**3.1.1. Process Ontology.** The proposed ontology of the process that is designed in this research is the result of the union of definition of the process, one from [6] and the other one from [18, 22]. In addition, some more concepts which are needed for the process to be successfully performed are considered in this research. By looking at the Garcia definition of the process, it can be understood that the process needs a context in order to show the society under discourse, a fact in order to find the wanted result that the process is looking for, and some activities in order to be performed. Also, by looking at the process definition by Davenport, it can be inferred that the process needs an objective to show the purpose of doing it, an owner to represent the person in charge of the process, and a recipient to show the target of the process. In addition to the above concepts three more concepts, namely, the possible fact, the performer, and the dependency, are introduced in this research. The performer is the one that has a duty to perform the process's activities. The dependency represents another process on which the main process depends. The possible fact gives information about the type and location of the fact (Figure 2).

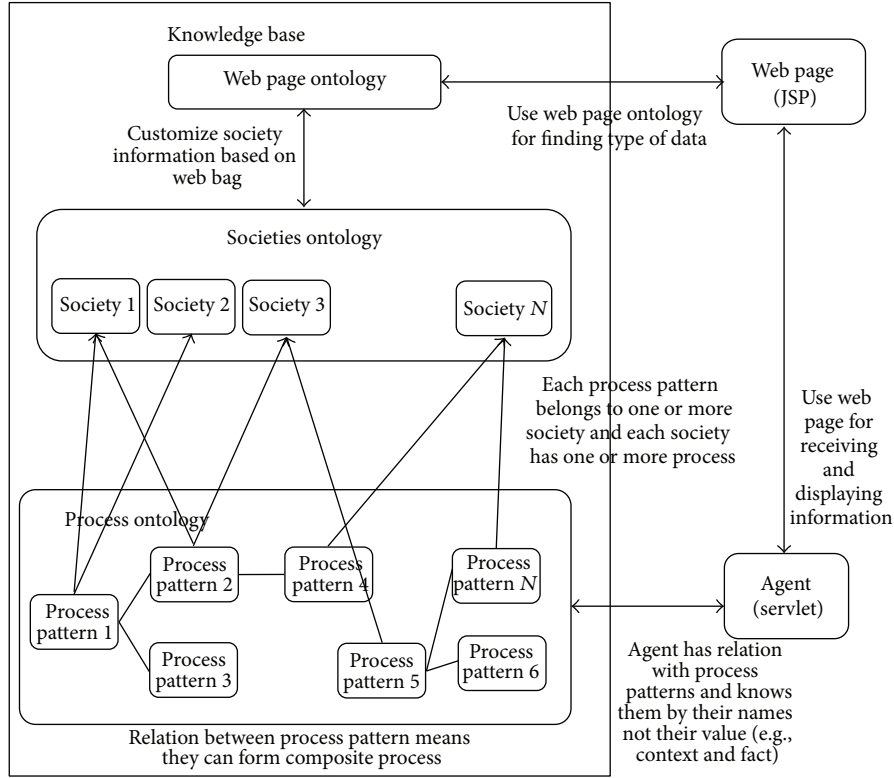


FIGURE 1: The proposed model.

*Definition 1* (recipient). One has

$$B = \{x \mid x \in (\{\text{Person}\} \sqcup \{\text{Process}\}), \exists p \in \{\text{Process}\}, \text{getBenefit}(x, p)\}, \quad (1)$$

where  $B$  is the recipient set and  $x$  is a variable which is of the process or the person type and can be a member of the recipient set if there is an instance of the process such as  $p$  such that the  $x$  gets benefit from the  $p$ . The recipient is the one that receives the result of the process. It can be either user or process. A process can be a recipient of another process which receives the result of that process. This way these two processes form a composite process. The process that its recipient and its dependency is the user is atomic process. Hence, the recipient of the main process in the composite process and the atomic process is the user. For example, the recipient of service providing process in bank login process is the user while this process constitutes a composite process with account finding and security code finding process.

*Definition 2* (owner). One has

$$W = \{x \mid x \in (\{\text{Person}\} \sqcup \{\text{Process}\}), \exists p \in \{\text{Process}\}, \text{initiator}(x, p)\}, \quad (2)$$

where  $W$  is the owner set and  $x$  is a variable which is of the process or the person type and can be a member of the owner set if there is an instance of the process such as  $p$ , such that the  $x$  initiates the  $p$ . The owner is the one that initiates the process

and is the reason for the existence of the process. The owner is either the user or the process. The owner is different from the recipient; for example, the owner can be a salesperson, while the recipient can be a customer in e-business society.

*Definition 3* (objective). One has

$$O = \{\text{Finding}\}, \quad (3)$$

where  $O$  is the objective set. Davenport considers this concept in order to show the context and purpose of the process. Since we define the context concept separately, the objective concept only shows the purpose of the process in our study. Besides, since we are dealing with a system, the objective of all processes is Finding.

*Definition 4* (context). One has

$$C = \{x \mid x \in \{\text{Society } I = 1 \dots N\}\}, \quad (4)$$

where  $C$  is the context set and  $x$  is a variable which can be any instance from society under discourse. The context of the process is equal to any object or value from the particular society result of which is supposed to be found among them. The context is analogous to the range of functions. The context can be directly displayed to the users once it consists of some individuals. However, in the cases where the context consists of literal, the agent will indirectly make the user understand what the context is. For example, the context of the security code finding process is all usernames and

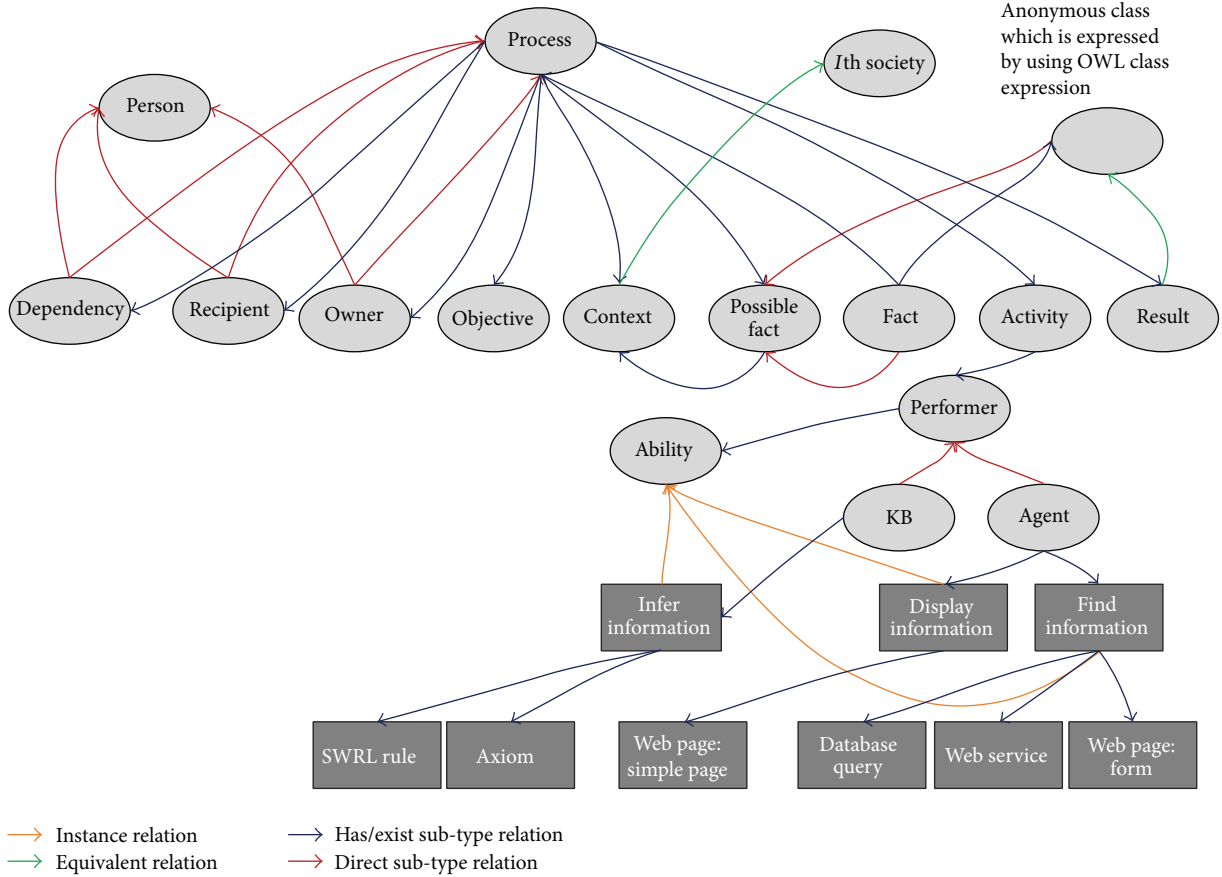


FIGURE 2: The process ontology.

passwords which indirectly are introduced to the user. In the composite process, the context of the subprocess is equal to the possible fact of the supper process.

*Definition 5* (possible fact). One has

$$D = \{x \mid x \in \{\exists Re \cdot C\}\}, \quad (5)$$

where  $D$  is the possible fact set and  $x$  is a variable which can be a member of the possible fact set if there is a property such as  $Re$  which connects it to some instances in the context  $C$ . In order for the process to find its result among the context's constituents, some facts are needed. Since these facts must be obtained through another process or user, it is necessary for the process to provide some information for them. The possible fact is a concept which is considered to provide information for the performer of the fact finding process. It is analogous to the domain of the function. Similarly to the context, the possible fact can be directly displayed to the users once it consists of some individuals. And, in the cases where it consists of literal, the agent will indirectly make the user understand what the possible fact is.

*Definition 6* (fact). One has

$$F = \{x \mid x \in \{D\}\}, \quad (6)$$

where  $F$  is the fact set and  $x$  is a variable which is a member of the  $D$  (possible fact) and can be a member of the fact set if it is chosen by the user or is the result of another process. Fact set consists of any information which is related to the context of the process and restricts it toward finding the results. Fact can be either object or literal; for example, an account is an object which is the fact for the service finding process, while the password is a literal which is the fact for the security finding process.

*Definition 7* (dependency). One has

$$N = \{x \mid x \in (\{Person\} \sqcup \{Process\}), \exists p \in \{Process\}, \text{depends}(p, x)\}, \quad (7)$$

where  $N$  is the dependency set and  $x$  is a variable which is of the process or the person type and can be a member of the dependency set if there is an instance of the process such as  $p$ , such that the  $p$  depends on the  $x$ . The dependency of the process is anything that can help process to find the value of its concepts. Since the process can find the value of its concepts through another process or the user, therefore, the dependency of the process is either the user or another process.

*Definition 8* (activity). One has

$$\begin{aligned}
 A = \{x \mid \exists b \in \{B\}, \exists w \in \{W\}, \exists o \in \{O\}, \exists c \in \{C\}, \\
 \exists d \in \{D\}, \exists f \in \{F\}, \exists n \in \{N\}, \\
 \exists r \in \{R\}, \text{valueFinder}(x, b, w, o, c, d, f, n, r), \\
 \text{hasPerformerOneOf}\{\text{KB}, \text{agent}, \text{user}\}\}, \quad (8)
 \end{aligned}$$

where  $A$  is the activity set and  $x$  is a variable which can be a member of the activity set if it can find value for the process's concepts and has a performer one of the KB, the agent, or the user. Since the activity is a complicated concept we refer to it in a separate part. But for now we can say an activity is anything that the KB, the agent, and the user do for finding the value of the concepts of the process. Therefore, the value of the activity concept cannot be stated explicitly except the last activity which says what to do with the result.

*Definition 9* (result). One has

$$R = \{x \mid x \in \{C\}, \exists f \in \{F\}, \text{hasRelation}(x, f)\}, \quad (9)$$

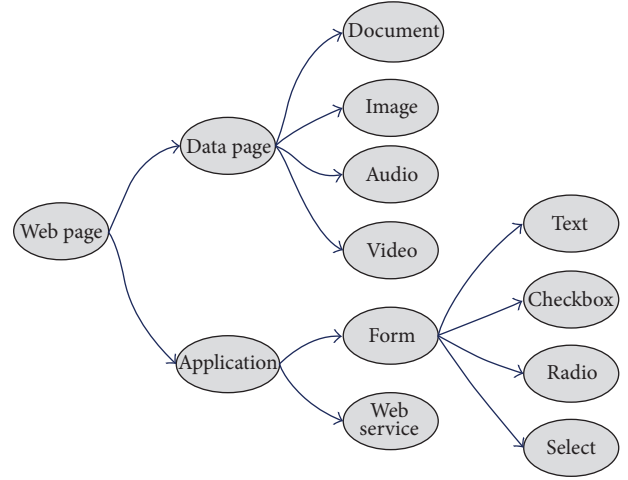
where  $R$  is the result set and  $x$  is a variable which is a member of the  $C$  (context) and can be a member of the result set if there is some facts such as  $F$  from the fact set, such that the  $x$  has a relation with the  $F$ . The result is the latest concept of the process which finds its value. The result's values are some of the things that are in the context of the process. If the context contains individuals, consequently the result value is the individual, or else if context refers to the literal, the result will have the literal as its value. The values of the result are obtained through the intersection of the context and some anonymous sets which are expressed by the fact.

*Definition 10* (process). One has

$$\begin{aligned}
 P = \{x \mid x \sqsubseteq \exists \text{Re} \\
 \cdot \{B\} \sqcup \{W\} \sqcup \{O\} \sqcup \{C\} \sqcup \{D\} \sqcup \{F\} \\
 \sqcup \{A\} \sqcup \{N\} \sqcup \{R\}\}, \quad (10)
 \end{aligned}$$

where  $P$  is the process set and  $x$  is a variable which can be a member of the process set if there are some relations such as  $\text{Re}$  which connects the  $x$  to some instances from the  $B$  (recipient),  $W$  (owner),  $O$  (objective),  $C$  (context),  $D$  (possible fact),  $F$  (fact),  $A$  (activity),  $N$  (dependency), and  $R$  (result). By referring to the above definition, we can define the process set as anything which has a relation with the recipient, the owner, the objective, the context, the possible fact, the fact, the activity, the dependency, and the result. In fact process is a function which has a domain (context), a range (possible fact), an input (fact), and an output (result).

**3.1.2. Web Page Ontology.** According to [23] a web page consists of images, documents, audios, videos, and animations such as flash and GIF files whose purpose is to give information about a particular subject to the recipients. Also, the web page can be used as an application for retrieving information



→ “has” relation

FIGURE 3: The web page ontology.

from the resources and transferring it to the databases or other parts of the application that need the information. Therefore, the web page contents can be categorized into two categories, namely, an application and a data page, at a very high level. Finding information by the web page can be done using three methods, asking of the end user, querying the database, and asking of other agents using web services (Figure 3). Asking of the end user needs to have a form which provides some entries in order for users to enter data in it. The entries of the form are categorized based on the owl definition of entities. If the needed data are of type literal, the form's entry will be text or select. And, if the needed data are of type individual, the form's entry will be checkboxes or radio buttons. Once the agent wants the web page to show information either as the result or the context of the process, the web page will look at its ontology to find out the type of information in order to display them by suitable content.

The web page, also, can have links to other pages which is considered as a service. However, this study does not consider them as the web page services, but as services of the society which the agent is performing its processes. As an example, the account query is a service of the bank which is displayed as a link in the web page. These services can be presented by using all the web pages' contents which are categorized as the data page.

*Definition 11* (web page). One has

$$WP = \{x \mid x \in \{C\} \sqcup \{D\} \sqcup \{R\}\}, \quad (11)$$

where  $WP$  is the web page set and  $x$  is a variable which is an instance of the  $C$  (context), or the  $D$  (possible fact), or the  $R$  (result). Since web pages' contents are based on the process, we define it by referring to the process concepts. The web page can contain anything which is in either the context, or the possible fact, or the result concept of the process.

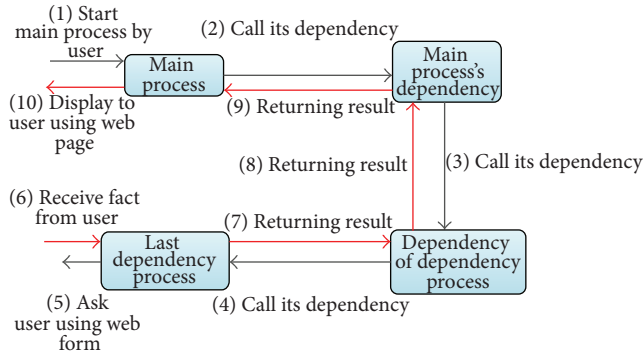


FIGURE 4: Composite process.

**3.2. Process Steps Analysis.** One process may find its result directly based on its initial information, which is called atomic process, or by getting help of other processes or the user, which is called composite process. The composite process will be started by the main process. The main process is that which its recipient is the user and its result will be displayed on the web page. The KB by using the axioms or SWRL rules will set all initial information for the main process. However, if the main process still needs more information, the KB will ask the agent to provide it by starting another process on which the main process depends. Therefore, it can be said that the main process will start its dependency process. Similar to the main process, the dependency process will call its own dependency for finding its needed information and so on. The value of the dependency concept of the latest process is equal to the user; therefore, systems use the form as a tool for getting information from the user. Once the latest process finds its result, it will return it back to the supper process by which it has been started. The agent will find the supper process by looking at the recipient of the current process. The recipient, in turn, will find its result by using the information that has already been received from its dependency. This procedure will continue till the main process finds its result and returns it to the user which is its recipient (Figure 4).

Based on the above description, the beginning and the end of the process can be clearly defined by using the recipient and the dependency concepts which are corresponding to the Davenport definition of the process. The order of the processes, also, can be defined by using these two concepts. The first process is the one which its recipient is the user and the latest process is the one which its dependency is the user or anything but the process.

### 3.3. Process's Activities

**3.3.1. Knowledge Base.** Based on the above definitions, the process is a set of concepts and the relation among them. The purpose of the process is to find the desired result based on the value of other concepts. Therefore, the result is the latest concept of the process which will find its value. Some activities are needed for giving value to the concepts. In fact, one process is needed for each concept in order to find its

value, and it can be said that the activity concept of the process refers to these subprocesses (Figure 5).

By looking at (Figure 5) it can be seen that there is a big problem with it. The problem is that if there is need to have some subprocesses for finding value for the concepts of the main process, therefore there is need to have many other processes for finding value for the concepts of the subprocesses. In order to solve this problem, almost all of the agent systems such as [4], and even though those that have KB such as [6] locate the knowledge of performing these activities inside the agents. This approach, however, prevents the agent from being independent. As can be seen in Figure 1, this research locates the knowledge of performing these activities inside the KB as a process pattern (Figure 6). Therefore, by doing so, the organizations are free to add or remove the process patterns from the KB by using the KB editor such as Protégé which is used in this research. Thus, the agent does not depend on the societies.

Assigning value to the process patterns will be done by either the KB or the agent. The approach is as follows.

- (1) The agent sets the process name.
- (2) The KB, based on the process name, will assign value to other concepts from societies as much as it can afford.
- (3) Once the KB is not able to find the value for the concepts, it will ask the agent to do that by defining the location where the agent can find the needed value.

We tried to infer value for the concepts of the process patterns by using the reasoning engine, such as Pellet or HermiT, as much as possible. The context of the process consists of all things that the result would be one of them. The context of the main process must be defined explicitly. Since the number of things that are considered as the context of the process are not defined and are changing continually (e.g., the number of accounts will change by creating or deactivating one account), we consider one SWRL rule for assigning context to the main process. The context of the subprocesses can be obtained through reasoning by using the possible fact of its supper process. The activity for finding the name for the process means calling one process pattern and setting it as the current process; thus, the system will understand which process is taking place. This activity is done by the agent and the user. The agent gets the name of the current process from the user and sets it in the KB. The fact concept needs to find its value either by the user or the subprocess. The fact finding activity is needed to be done by the agent. However, the KB will tell the agent what the possible facts are and where they can be possibly found. If the type of the possible fact is an individual, the KB will find the possible fact by finding any things which have relation to the context. But, unfortunately, it is not possible to find the possible fact which is of type literal by using reasoning engine. However, the KB will tell the agent which type of literal the possible fact is. The agent by finding the fact will construct an axiom and set it as the result class definition. The KB by using the reasoning will find result's value based on the result class axiom.



FIGURE 5: Process activity.

The dependency concept of the main process refers to the process which is used for finding the fact of the main process. Therefore, the dependency concept is considered as a means to establish communication between the supper process and the subprocess. The last activity is the delivery of the process's result to its recipient which will be done by the agent. The agent will understand what to do with the result by checking the recipient of the process. The recipient of most of the processes will be obtained by using the reasoning engine. If the recipient is the user, the agent will ask the web page to display the result to the user. The web page activity is a concept which is considered to tell the web page to show the result to the user. This value will be assigned to the web page activity concept using one SWRL rule. However, if the recipient of the process is the supper process, the agent will set the supper process as the current process and set the result as the fact for the supper process.

It can be said that each process pattern is a function which we moved from the inside of the agent to the KB. The context of the process pattern can be considered as the range of the function, the possible fact as the domain, the fact as the input argument, and the result as the output. By doing so, we provide flexibility for the system in terms of adding or removing the functions to or from the KB in order to bring the independency to the agent.

3.3.2. *Agent.* As mentioned before, a process needs some subprocesses to be performed. There are times that the subprocesses also need another subprocess to find their results. And, once the latest process finds its result, it needs to give it to its recipient. The recipient considers the received result as its fact and, then, by using that, will find its own result. And this procedure must continue till the main process finds its result and gives it to its recipient which is the user.



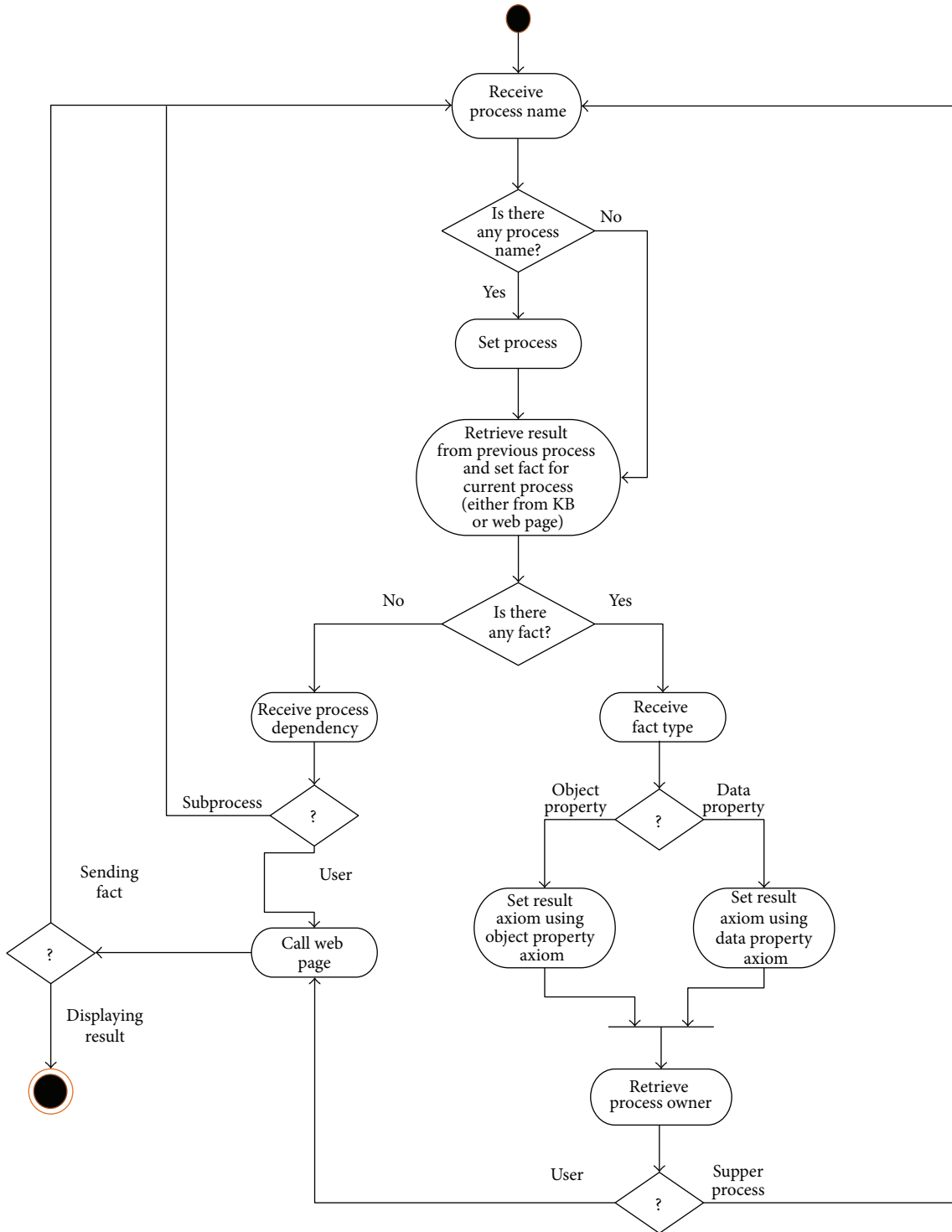


FIGURE 6: Agent activity diagram.

In order for these activities to be performed, besides the KB another performer is needed to do some of the activities which the KB is not able to do. Therefore, one agent who has knowledge of the process concepts is introduced. OWL\_API

version 3 is utilized in order to relate the agent with the KB. The agent will be called in order to do two activities of the process's activities, namely, starting the process and finding the fact of the process. The agent starts its task by checking

the process name (Figure 6). The existence of the process name means starting a new process, or else it means the agent has received the fact and needs to set it in the KB.

The process name will be defined either by the user by using the web page or by another process using the dependency and the recipient concept. The fact comes either from the web page or the previous process. The fact from the web page means the result of the subprocess that is obtained by asking the user, and fact from the KB means the result of the subprocess that is obtained by the reasoning engine. By receiving the facts, the agent needs to construct the result axiom and set it in the KB. Since the type of the fact may vary from the individual to the data value, two separate functions are defined. After setting the result axiom, the KB will find the result. The agent checks the recipient of the process to see what it needs to do for the next step. If the recipient is the user, the result will be displayed to the user by using the web page, or else if the recipient of the process is another process, the result will be set as the fact of that process.

**3.3.3. Web Page.** The agent has two purposes of calling the web page. The one is getting the fact from the end user, and the other is displaying the result to the user. Hence, the web page needs to know what reason it has been called for. In order to give this knowledge to the web page, the web page activity concept is defined. The web page activity concept can possess two values either "Display\_To\_User" or "Null." If the web page activity is equal to null, it means that the reason it has been called for is finding the fact; therefore, the web page will start the respective process by obtaining its name from the KB (Figure 7). Then, it will find the context of the process. Since the context of the subprocess is the possible fact of the supper process, the web page will look for the context which the possible fact of the supper process refers to; then by using its form it will get the fact from the user. If the web page activity is equal to the "Display\_To\_User," it means that the reason that the web page has been called for is displaying the result; therefore, the web page uses the data page contents to show the result to the end user.

Once the reason for calling the web page is finding the fact, the web page needs to know the type of the fact (since the type of the fact is the same as the possible fact and the possible fact of the supper process is equivalent to the context of the subprocess, therefore, the type of the context and the fact are the same in the security code finding process). For this purpose, based on the categorization of the property by OWL-Ontology which are the data property and the object property, the context will be divided into two categories, namely, the object or the literal. If the type of the context is the object, the context class will be equal to the class that contained those objects. Therefore, the web page by using DL Query will retrieve them. However, if the type of context is the literal or any other type which is a subtype of the literal, the context class cannot contain them, but it will indirectly refer to that literal. The web page needs to give some clues to the user in order to make him understand what the context is and what type of fact you need to provide in order to find the result. For example, if the context of the security code

finding process refers to "securityCode" the web page needs to display two entries, one for receiving the username and the other one for receiving password to the user in order to make him understand that the needed facts are the username and the password. The web page will do that by finding the name of the data property that connects the possible fact and the context of the supper process. For example, "hasUsername" and "hasPassword" are two data properties which connect the context and the possible fact of the account finding process. Then, the web page by using the substring method of the string class will remove the "has" from them and display them to the user. Finding data property will be done by SPARQL query. Once the web page found the context of the process, it needs to know how the context must be displayed. The literals always will be displayed using the text input entry. However, the type objects will be obtained using web page ontology.

#### 4. Experimental Setting and Evaluation

The objective of this evaluation is to determine the applicability and the quality of the proposed system in the performing processes from the different societies. Since the proposed system is modular, by checking the quality of each module we can find the quality of the system. The cohesion and the coupling are two measures for evaluating software that has been investigated for many years in the software engineering society [24]. The cohesion and the coupling measure the degree to which the elements of a module are either related to each other or to the external elements. In other words it can be said that the cohesion and the coupling investigate the knowledge of a module to find the degree to which the concepts of that knowledge are related to each other or the degree to which the concepts of that knowledge are related to the concepts of the knowledge of the external modules. Since the process ontology can be considered as the knowledge of the agent, we need to investigate this ontology from the cohesion and the coupling points of view.

In order to evaluate the proposed system from the applicability perspective and also determining the manageability and the automatability of the processes, we define three societies and seven processes. We will logically, by inputting data and tracking the process for finding the desired output, investigate the system. In order to do so, we consider the bank, Facebook, and a mobile sales company as societies. We consider two login processes from Facebook and the bank and define two ontologies for these two societies and add to the KB. We define three process patterns, namely, the service finding process, the account finding process, and the security code finding process. In the mobile sales company, we define a very brief ontology for the mobile and the customer and also four process patterns, namely, the e-mail finding process pattern, the potential customer finding process pattern, the mobile characteristic process pattern, and the mobile finding process pattern. Then, we track the service finding process from the bank and Facebook societies and the e-mail finding process from mobile sales company society in order to show the applicability of the system in these two societies, as well

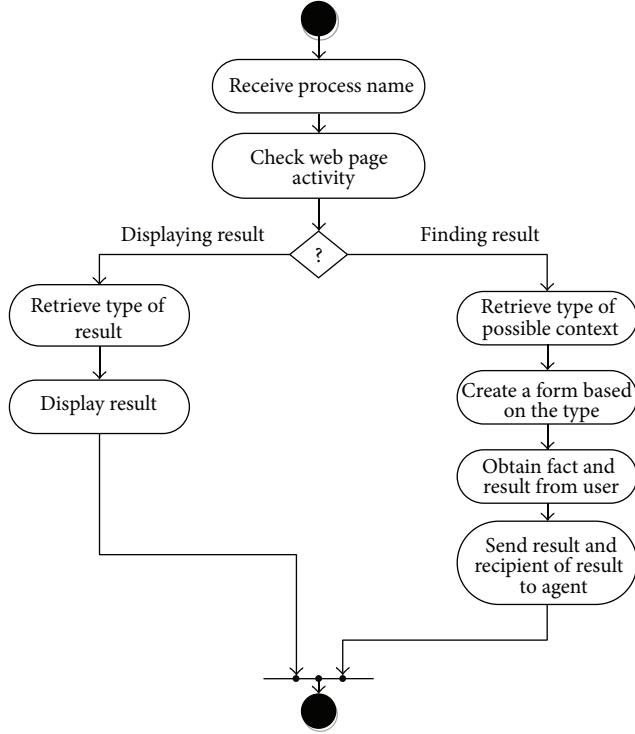


FIGURE 7: Web page activity diagram.

as the manageability and the automatability of the processes by system (Table 3).

#### 4.1. Modularity Evaluation of the Ontology

**4.1.1. Metrics Description.** Two numeric metrics for evaluating modularity of ontology model, namely, the cohesion and the coupling, are introduced by [25]. An ontology module is more understandable than another if it has higher value for the cohesion and less value for the coupling than another. According to Ensan, a cohesive ontology is the one which its conveyed information are all describing a very specific domain of discourse. In other words, the dependency degree among the concepts of an ontology module represent its cohesion. For example,  $OM = \{A, B\}$  is not cohesive because there is not any relation between its concepts, whereas,  $OM = \{A \subseteq B\}$  is highly cohesive because there is a subsumption relation between its concepts. The ontology module which has the least dependency on external information is considered lowly coupled. In other words, the dependency degree among the concepts of an ontology module to another ontology module represents its couplingness. For example, consider  $OM = \{A \subseteq B\}$  in which  $B$  is the concept of the external ontology module such as  $OM'$ . In this case,  $OM$  is considered highly coupled on  $OM'$  since  $A$  is dependent on  $B$ .

Before referring to the cohesion and the coupling, it is necessary to know what the dependency is. According to Ensan, the dependency of a concept or a role in an ontology is determined by checking its semantics. The semantics of a concept or a role will be interpreted by defining its domain. If

the domain of a concept or a role is affected by others, while the domain of the others is set up, we say the semantics of the concept or the role depend on the others. Ensan defines two types of dependencies, namely, the strong dependency and the moderate dependency. If the other concepts or roles cause the domain of a concept or a role becomes equivalent to the top or the bottom concepts (OWL: Thing, OWL: Nothing, OWL: topObjectProperty, OWL: bottomObjectProperty, OWL: topDataProperty, and OWL: bottomDataProperty), we say the concept or the role strongly depends on the other concepts or roles. For example, if  $OM = \{A \subseteq B\} \not\models A \equiv \perp$  and  $OM - B = \{A \subseteq B, B \subseteq \perp\} \models A \equiv \perp$ , then we say that  $A$  is strongly dependent on  $B$ . The strong dependency of  $A$  is shown by  $SDepM(A)w$  (which is equal to all the concepts in  $M$  on which  $A$  strongly depends). If the domain of a concept is dependent on two other concepts, we call it the moderate dependency of degree one. And if the domain of a concept is dependent on three other concepts, we call it the moderate dependency of degree two and so on. In a formal definition,  $A$  moderately depends on  $B$  if  $A$  does not strongly depend on  $B$  and  $SDepM(A) \neq SDepM - B(A)$  or  $SDepM(A) \neq SDepM + B(A)$ . For example, in  $OM = \{A \subseteq B \sqcup C\}$ ,  $SDepM(A) = \emptyset$  and  $SDepM - B(A) = \{C\}$ ,  $A$  has first-degree moderate dependency on  $B$ . The moderate dependency of  $A$  is shown by  $MDepM(A)$  which is equal to all concepts in  $M$  on which  $A$  moderately depends.

$SDepLoc(M)$  represents the strong dependency among the concepts of one ontology module.  $SDepExt(M)$  represents the strong dependency of the concepts of the local ontology module on the external ontology module. Similarly,  $MDepLoc(M)$  represents the moderate dependency among the concepts of one ontology module.  $MDepExt(M)$  represents the moderate dependency of the concepts of the local ontology module on the external ontology module. NSLD (Number of Strong Local Dependency) and NMLD (Number of Moderate Local Dependency) are two metrics for measuring the cohesion of ontology module.  $NSLD(M) = |SDepLoc(M)|$  and  $NMLD(M) = |MDepLoc(M)|$ . MLD (Maximum Local Dependency) represents the number of all potential local dependencies and is obtained through  $MLD(M) = n * (n - 1)$  in which  $n$  represents the number of the symbol of the local ontology module  $n = |Loc(M)|$ . By having defined these metrics, the cohesion of an ontology module will be obtained as follows:

$$\begin{aligned}
 COH(M) &= \begin{cases} 1, & \text{if } |Loc(M)| = 1 \\ \frac{\gamma * NSLD + \delta * NMLD}{(\gamma + \delta) * MLD}, & \text{if } |Loc(M)| > 1. \end{cases} \quad (12)
 \end{aligned}$$

Similarly, we can find the coupling of an ontology module using the following function:

$$\begin{aligned}
 COP(M) &= \begin{cases} 0, & \text{if } |Loc(M)| = 0 \\ \frac{\gamma * NSED + \delta * NMED}{(\gamma + \delta) * (|Loc(M)| * |Ext(M)|)}, & \text{if } |Loc(M)| > 0. \end{cases} \quad (13)
 \end{aligned}$$

NSED (Number of Strong External Dependency) represents the strong dependency of the concepts of the local ontology module on the external concepts. NMED (Number of Moderate External Dependency) represents the moderate dependency of the concepts of the local ontology module on the external concepts:  $NSLE(M) = |SDepExt(M)|$  and  $NMED(M) = |MDepExt(M)|$ .  $(|Loc(M)| * |Ext(M)|)$  represents the number of all potential external dependencies.  $\gamma$  and  $\delta$  represent the impact of the strong and the moderate dependencies on the cohesion and the coupling.

*4.1.2. Process Ontology Evaluation.* As mentioned in Section 3, our proposed ontology for the process is defined as follows:

$$\begin{aligned}
 PO = \{ & (B \subseteq (Pesron \sqcup P)), (W \subseteq (Pesron \sqcup P)), O, \\
 & (C \equiv S_i), (D \subseteq \exists Re \cdot C), (F \subseteq D), \\
 & (N \subseteq (Pesron \sqcup P)), A, (R \subseteq (C \sqcap \exists Re \cdot F)), \\
 & (P \subseteq \exists Re \cdot B, \exists Re \cdot W, \exists Re \cdot O, \exists Re \cdot C, \exists Re \cdot D, \\
 & \exists Re \cdot F, \exists Re \cdot N, \exists Re \cdot A, \exists Re \cdot R) \}.
 \end{aligned} \tag{14}$$

The issue that we encounter is the evaluation of the size of the external ontologies. Since  $S_i$  is variable and represents different societies based on the usage of the system, we are not able to find its size. Hence, we consider the size of  $S_i$  as the average of the size of the three ontologies which we consider in evaluating the applicability of the system, namely, Facebook and the bank login system and the mobile sales company. We consider that  $\gamma$  equals 2 and  $\delta$  equals 1:

$$\begin{aligned}
 COH(P) &= \frac{(2 * 33) + (1 * 0)}{(2 + 1) * 90} = 0.24 \\
 COP(P) &= \frac{(2 * 4) + (1 * 6)}{(2 + 1) * 100} = 0.05.
 \end{aligned} \tag{15}$$

The degree of the cohesion and the coupling varies between 0.00 and 1.00, such that the highly cohesive ontology is the one whose cohesion is near to 1 and the lowly coupling ontology is the one whose coupling is near to 0.00. Figure 8 shows the result of the quality evaluation of the process ontology or, in other words, the agent knowledge. As can be seen the cohesion of the process ontology is 0.24 of 1.00, while its coupling is 0.05 of 1.00 which will definitely be less than this once the external ontology is completely defined, not very briefly as we did. Although the cohesion of the process ontology is not very high, it is much higher, more than four times than coupling of the process ontology which means the proposed ontology is focused on solving a particular problem. The reason that cohesion is not very high is the activity concept. Although it has a relation with all the concepts and the performers, we are not able to show its relation with other concepts.

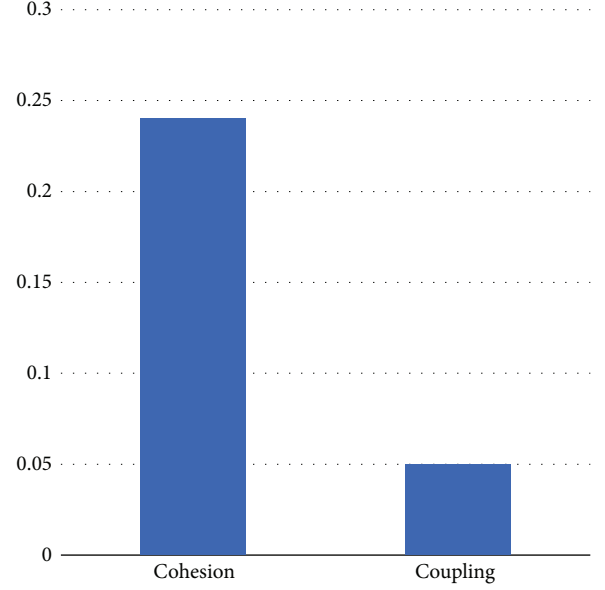


FIGURE 8: Quality result.

## 4.2. Applicability Evaluation

*4.2.1. Login Process.* By comparing two login processes, one from Facebook and one from the bank, it can be inferred that the steps of the login process in both of them are the same. The main purpose of these two processes is preparing a web page along with some service or information to users based on their identities. These two processes need some subprocesses to meet their purposes. Since our intention was to track a process at a very detailed level, two ontologies, at a very general level, were designed in order to evaluate our proposed system. Using these two ontologies we track a process which its purpose is finding some services for a particular user. The bank set consists of accounts. The account is divided, at a very high level, into two categories, the deposit account and the saving account. Each account is defined by some services that it offers and is secured by the security code. The account also has an asset and an owner. The asset is the existing money in the account. The owner of the account is the one whose nickname is equal to the user name of the account.

Therefore, the account set consists of the service, the security code, the asset, and the owner which is of type person (Figure 9). Since the services are countable and are already defined, they are considered as the object, whereas the username, the password, and the asset are data values. In order for the username and the password to be distinguished from other data type, a data type with the name of security is introduced.

Similarly to the bank ontology, Facebook ontology consists of the account, the service, the owner, the security code, and some information of the string type. The web page will display the services of both of them as a link.

*4.2.2. Service Finding Process.* The service finding process will be started once the user clicks on the login button.

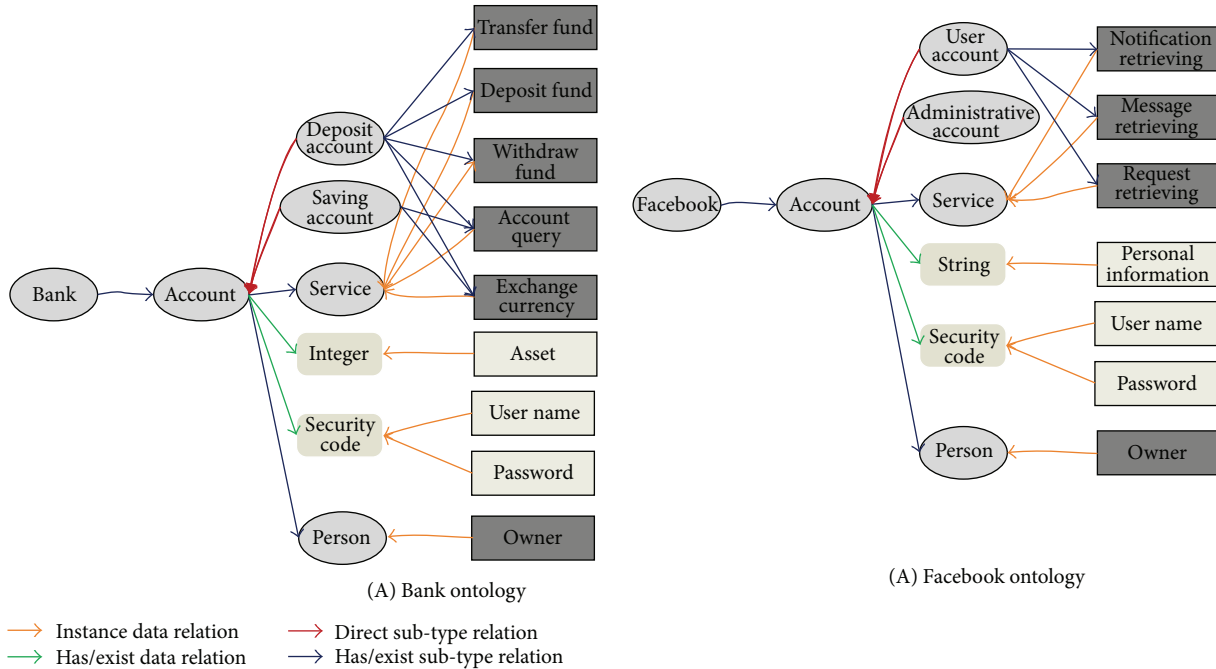


FIGURE 9: Bank and Facebook ontology.

The agent will set this process as the current process in the KB (Figure 10). The KB by using the reasoning engine assigns all the services to the service finding process as its context. Also, the KB by using the context property defines all the accounts as the possible fact for this process. However, since the fact of the process still is not defined, the KB is not able to find the result of this process. The agent by using the dependency concept will understand that the service finding process depends on the account finding process; therefore, the service finding process will be kept aside and the account finding process will be started. The KB will set the recipients of the account finding process with the service finding process and set the possible fact of the service finding process as the context for the account finding process which is all the accounts. Logically, the KB should set all the security codes, all the usernames and the passwords, as the possible fact of the account finding process. However, since the type of all the usernames and the passwords is of the “security,” which is defined by the researcher as a subtype of the literal, and literal cannot be inferred using reasoning engine, the KB is not able to display them. However, the KB will provide two clues, namely, “hasPossibleDataFact” which is data property and “Security” which is data value, for the web page to find the possible fact (Figure 6). The web page by using SPARQL asks the KB to provide anything that is a subtype of the “hasPossibleDataFact” and has the value of the type “Security.”

Since the fact of the account finding process is not defined, this process cannot have the result. The agent by using the dependency concept of the account finding process will understand that this process needs the security finding process in order to find its fact. Therefore, the account finding

process will be kept aside and the security code finding process will be started.

The recipient of the security code finding process will be set with the account finding process, and the context of it will be set with the possible fact of the account finding process. Since this process is dependent on the user, performing of this process will be assigned to the web page.

The web page by finding “hasUsername” and “hasPassword” data properties and manipulating them will display “UserName” and “Password” to the user in order to make him understand that the needed information is the user name and the password. By receiving the username and the password, the security code finding process will find its result. The agent by checking the recipient concepts of the security code finding process will find that the account finding process needs this result; therefore, it will set this result as the fact for the account finding process. Since the account finding process has found its fact, the KB by using the reasoning engine will find the result of this process which is “MahmoodAccount”.

The agent will find the recipient of the account finding process which is the service finding process; therefore, it will set the result of the account finding process as the fact for the service finding process. The KB, again, by using the reasoning will find the result of the service finding process. Since the recipient of the service finding process is the user, the agent will call the web page for displaying the result to the user. Since the user has the saving account from the bank, two services, namely, the account query and the exchange currency, will be offered to him.

The similar processes will be performed for finding some of the services by the Facebook users based on the type of the account that are offered to them.

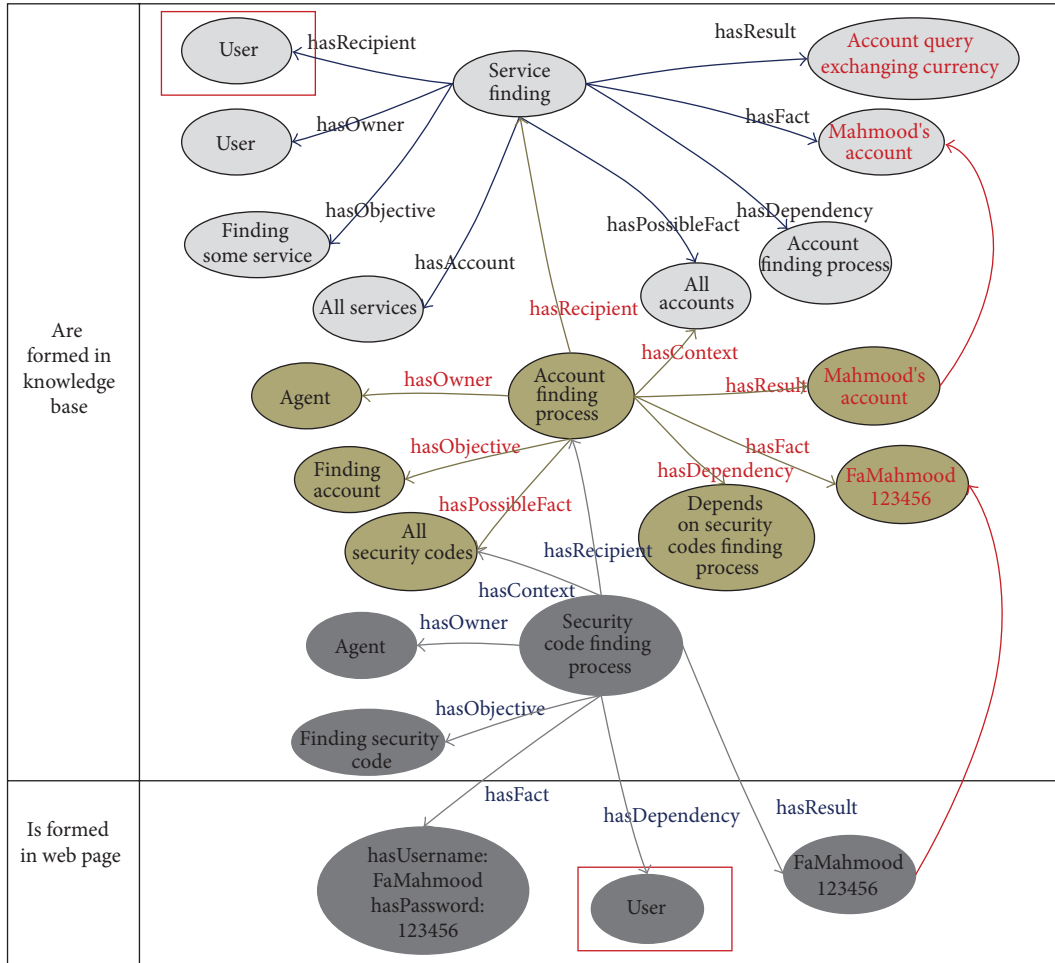


FIGURE 10: Service finding process steps.

4.2.3. *E-Mail Finding Process.* The third society which we consider to evaluate the applicability of our proposed system is the mobile sales company. This company, in order to find its potential customers, needs to have knowledge of its products and knowledge of the relationship between its products and potential customers. Once the potential customers are found, the company needs to find their e-mails in order to send them some advertisements. Therefore, the first objective is finding potential customers e-mail (Figure 11).

**5. Discussion**

Ontology process with clearly defined concepts was proposed in this research as a means of addressing the problem of the dependency of the systems on the societies. The activities that are needed for the process to be performed, which have not been defined in the previous researches, were clearly defined in this research. The performer, namely, the agent and the KB, of performing the activities and the means by which they can perform their tasks were clearly defined. The web page, the database query, and the web service were considered as tools for the agent to be able to perform its tasks, and the axiom and SWRL were considered as

tools for enabling the KB to perform its tasks. Since the process's concepts are clearly defined inside the KB, the only thing that organizations need to do for using the system is instantiating the process's concepts according to their organizations' societies. The proposed system was evaluated in Section 4 from the quality and the applicability points of view. From the quality perspective, we evaluate our proposed ontology of the process by using two measures, namely, the cohesion and the coupling. The result showed that the cohesion is more than four times higher than the coupling which implies that the knowledge of the agent is focused on solving a particular problem (Figure 11).

The proposed system was evaluated from the society-independency point of view by proving its applicability in three different societies. In order to evaluate the proposed system from the manageability and the automatability perspectives, we distinguish the managed processes from the automated processes by their initiator and dependency. If one process can be performed and find its result, regardless of its performer, then it is considered as the managed process, whereas, if one process can be performed by the system, without intervention of the user, it is considered as the automated process. In other words, the automated process is

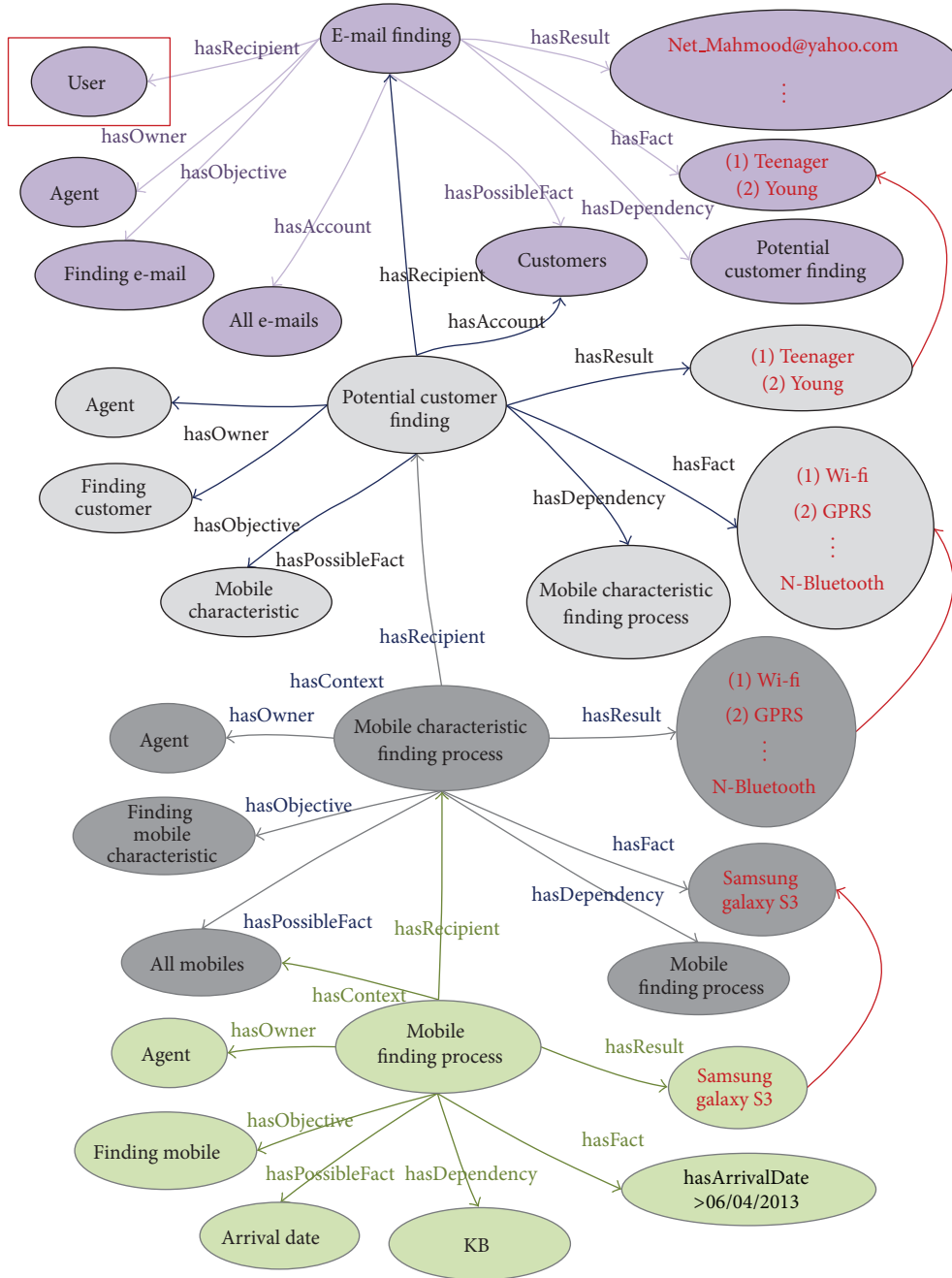


FIGURE 11: E-mail finding process step.

the one that is initiated by the agent and does not depend on the user for finding its fact. There is a material conditional (material implication) between the automated and the managed process, such that if the process is automated then it is also managed, but not vice versa (The automated process  $\rightarrow$  The managed process). Among the seven different processes which were instantiated from the process ontology (Figure 2) and performed in the three societies, the service finding process was initiated by the user, and the security code finding process was dependent on the user for finding its fact; therefore, these two processes are considered as only

the managed processes. The others all were initiated by the agent and were not dependent on the user for finding their facts; therefore, they are considered as the automated processes. Since the automated process is also the managed process, all of the seven processes were managed by the system, and five of them were automated (Table 2).

## 6. Conclusion

Management and automation of the processes are essential for the organizations in order to achieve their objectives. IT





- [2] J. J. Jung, "Contextual synchronization for efficient social collaborations in enterprise computing: a Case Study on TweetPulse," *Concurrent Engineering-Research and Applications*, vol. 21, no. 3, pp. 209–216, 2013.
- [3] J. J. Jung, "Reusing ontology mappings for query routing in semantic peer-to-peer environment," *Information Sciences*, vol. 180, no. 17, pp. 3248–3257, 2010.
- [4] Q. Liu, S. X. Sun, H. Wang, and J. Zhao, "A multi-agent based system for e-procurement exception management," *Knowledge-Based Systems*, vol. 24, no. 1, pp. 49–57, 2011.
- [5] Y.-S. Chang, C.-T. Yang, and Y.-C. Luo, "An ontology based agent generation for information retrieval on cloud environment," *Journal of Universal Computer Science*, vol. 17, no. 8, pp. 1135–1160, 2011.
- [6] A. Garcia-Crespo, B. Ruiz-Mezcua, J. L. Lopez-Cuadrado, and I. Gonzalez-Carrasco, "Semantic model for knowledge representation in e-business," *Knowledge-Based Systems*, vol. 24, no. 2, pp. 282–296, 2011.
- [7] J. J. Jung, "Evolutionary approach for semantic-based query sampling in large-scale information sources," *Information Sciences*, vol. 182, no. 1, pp. 30–39, 2012.
- [8] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing?" *International Journal on Human-Computer Studies*, vol. 43, no. 5-6, pp. 907–928, 1995.
- [9] Y. S. Chang, T. Y. Juang, C. H. Chang, and J. S. Yen, "Integrating intelligent agent and ontology for services discovery on cloud environment," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC '12)*, pp. 3209–3214, Seoul, Korea, October 2012.
- [10] J. Dang, A. Hedayati, K. Hampel, and C. Toklu, "An ontological knowledge framework for adaptive medical workflow," *Journal of Biomedical Informatics*, vol. 41, no. 5, pp. 829–836, 2008.
- [11] J. J. Jung, "Service chain-based business alliance formation in service-oriented architecture," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2206–2211, 2011.
- [12] B. Vesin, M. Ivanović, A. KlačNja-MilićEvić, and Z. Budimac, "Protus 2.0: ontology-based semantic recommendation in programming tutoring system," *Expert Systems with Applications*, vol. 39, no. 15, pp. 12229–12246, 2012.
- [13] J. J. Jung, "Semantic optimization of query transformation in a large-scale peer-to-peer network," *Neurocomputing*, vol. 88, pp. 36–41, 2012.
- [14] K. A. Delic and J. A. Riley, "Enterprise knowledge clouds: next generation km systems?" in *Proceedings of the International Conference on Information, Process, and Knowledge Management (eKNOW '09)*, pp. 49–53, February 2009.
- [15] N. Villanueva-Rosales and M. Dumontier, "yOWL: an ontology-driven knowledge base for yeast biologists," *Journal of Biomedical Informatics*, vol. 41, no. 5, pp. 779–789, 2008.
- [16] Y. Liu and N. Guihua, "Implementation of supply chains coordination using semantic web service composition," in *Proceedings of the IEEE International Conference on Service-Oriented Computing and Applications (SOCA '07)*, pp. 249–253, June 2007.
- [17] J. J. Jung, "Cross-lingual query expansion in multilingual folksonomies: a Case Study on Flickr," *Knowledge-Based Systems*, vol. 42, pp. 60–67, 2013.
- [18] T. H. Davenport, *Process Innovation*, Harvard Business School Press, Boston, Mass, USA, 1993.
- [19] K. H. Han and J. W. Park, "Process-centered knowledge model and enterprise ontology for the development of knowledge management system," *Expert Systems with Applications*, vol. 36, no. 4, pp. 7441–7447, 2009.
- [20] R. Woitsch and D. Karagiannis, "Process oriented knowledge management: a service based approach," *Journal of Universal Computer Science*, vol. 11, no. 4, pp. 565–588, 2005.
- [21] D. Cerri, "Towards knowledge in the cloud," in *On the Move to Meaningful Internet Systems*, vol. 5333 of *Lecture Notes in Computer Science*, pp. 986–995, 2008.
- [22] J. J. Jung, "ContextGrid: a contextual mashup-based collaborative browsing system," *Information Systems Frontiers*, vol. 14, no. 4, pp. 953–961, 2012.
- [23] P. Morville and L. Rosenfeld, *Information Architecture for the World Wide Web*, O'Reilly Media, 1998.
- [24] S. R. Chidamber and C. F. Kemerer, "Metrics suite for object oriented design," *IEEE Transactions on Software Engineering*, vol. 20, no. 6, pp. 476–493, 1994.
- [25] F. Ensan and W. Du, "A semantic metrics suite for evaluating modular ontologies," *Information Systems*, vol. 38, no. 5, pp. 745–770, 2013.
- [26] A. R. Basariya and O. S. J. Nisha, "Semantic based query optimization using ontology for knowledge management," in *Proceedings of the International Conference on Advances in Engineering, Science and Management (ICAESM '12)*, pp. 332–336, March 2012.