B.H. Far's Research
© B.H. Far
far@ucalgary.ca
http://www.enel.ucalgary.ca/People/far

Theme 11: Intelligent Software Measurement System (ISMS)

Abstract

Intelligent Software Measurement System (ISMS) is an agent-based system that is able to automatically produce a software measurement implementation plan based on users' initial business or measurement goal(s). The ISMS is composed of a Personal Assistant agent (PA) as the client interface, and a cluster of Expert Assistant agents (EA). The PAs provide users with simple business goals template for selection, and then through communication with the EAs they proceed to the goal(s). A central component of EA is the Knowledge Base (KB). We present a step by step methodology for building the knowledge base for ISMS. Unlike conventional knowledge based systems, the knowledge base in ISMS is composed of a series of relational tables, weights and factual rules. Automated learning and update mechanisms are also provided for performance improvement.

1. Introduction

Software measurement, in order to be effective, must be [4] focused on specific goals; applied to all life-cycle products, processes and resources; and interpreted based on characterization and understanding of the organizational context, environment and goals [4].

The Goal-Question-Metric (GQM) was developed in response to the need for a goal-oriented approach that would support the software measurement. A GQM model starts with a measurement goal. The goal is refined into several questions, and then each question is refined again into metrics.

For measurement to be cost effective, it must be designed and targeted to support the Business Goals of the organization [1]. Goal-Driven software measurement, which is a 10 steps GQM-centric process, is able to help users find and define software measurement that directly supports the organization's business goals. These measurements are traceable back to the business goals, so that the data-collection activities are better focused on users' intended objectives.

The 10 steps of Goal-Driven process are organized into three sets of activity (Fig. 1.): identifying goals, defining indicators and data needed to produce them, and creating an action plan to guide the implementation [6]. The first set of activities translates a business goal to quantifiable measurement goals. The second set produces a measurement plan that is aligned with the organizations' business processes. Successful design and implementation of this plan depends on experience of the measurement team. Identifying relevant questions, grouping questions, refining subgoals, identifying entities and attributes, formalizing measurement goal, and defining quantifiable questions and related indicators [1] all require measurement team members' knowledge and experience.

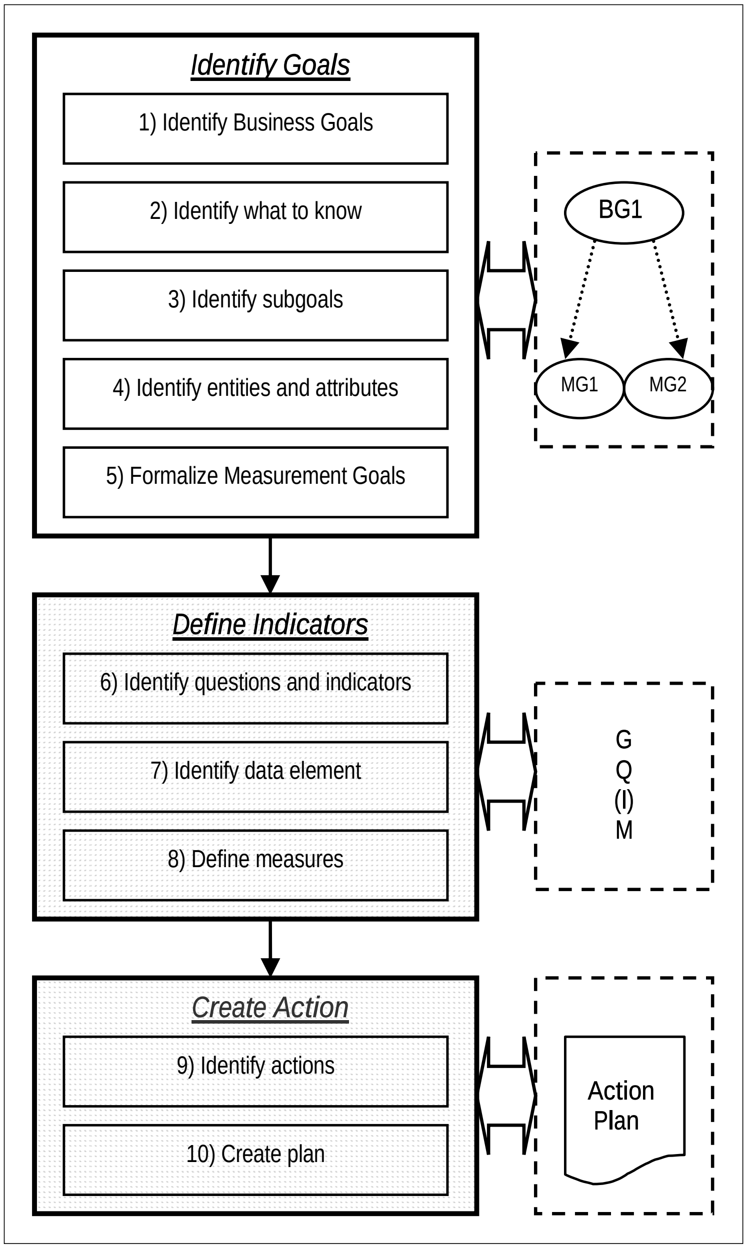

Fig. 1. Goal-Driven software measurement process.

The Intelligent Software Measurement System (ISMS) is developed following the Goal-Driven Software Measurement Process. In the ISMS project we automate the 10 steps of the process. The main development tasks of ISMS are eliciting the knowledge and experience from software measurement experts, representing it in a flexible yet well-structured way, and building a knowledge base infrastructure for the system. Using the knowledge infrastructure, ISMS provides users with a series of interactive screens and views which guide them through the Goal-Driven process. Below we present a methodology for building the knowledge base of ISMS, which helps users obtain measurement goals from business goals.

2. Knowledge Representation

2.1 Domain analysis

In order to build a knowledge base for intelligent systems, firstly we must understand the domain well enough to confirm what entities and facts are necessary to be discussed and what material can be ignored [2]. In this case, within the domain of goal-driven software measurement, our main purpose is finding out measurement goals to support the corresponding business goal. We follow a five-step procedure for implementing this target.

With the business goal identified, the second step is identifying the necessary information that related to the activities in order to achieve the business goal. This is done by asking entity-questions. After that, based on the pre-defined grouping keywords, above questions are grouped under several categories, which are manageable subgoals. The next step is to use the prioritized subgoals and entity-questions to refine entities and attributes. Finally, by collecting this information with purpose, perspective and environment elements, we can formulize measurement goals to support our original business goal.

Table 1. Objectives and dimensions

	Objectives	Dimensions
1	Business goal	- Business goal
2	Entity-Question list	- Entity-Questions
3	Subgoal	- Grouping keywords - Subgoals
4	Entity/Attribute	- Entities - Attributes
5	Complete measurement goal	- Purpose - Perspective - Environment

By the analysis of the process, all the elements needed to achieve the goal can be identified. For this, the initial input is business goal, and the terminal output is measurement goals. Besides the input and output, there are a set of objectives related to every step. Each objective is described by one or more dimensions shown in Table 1.

2.2 Organizing knowledge

We have broken the domain into five smaller clusters of knowledge.

- **Business goal:** The target of an organization, that the organization is able to reach [1].
- **Entity-Question list:** A list includes important process entities and a set of questions related to entities for supporting the business goal [1].
- **Subgoal:** The manageable convergent category title, by which entity-questions can be grouped and then the issue they address can be identified. The grouping keywords are the index of grouping procedure.
- **Entity/Attribute:** The entities that are implicit in the questions and the pertinent attributes related to objectives. The pairs of entity and attributes are potential objects of measurement.
- **Measurement goal:** A five dimensions goal based on the general corporate objectives, which includes object of interest (entity), quality focus (attribute), purpose, perspective, and environment description [3].

2.3 Refining relationship

We used the reverse engineering technique to translate business goal into the five tuples of the measurement goal from the original business goal. Formal measurement goals come from integration of entities, attributes and other information. The necessary entity-attribute pairs are identified from entity-questions, and these questions have been grouped under some prioritized and elected subgoals. The goal of setting entity-question list is to find associated entities in order to support business goals.

In the five dimensions of a measurement goal, the purpose, perspective, and environment context are easily figured out by user. The remaining two, entity and attribute, both are included in entity-question list. Hence, these activities form a "question-centric" process. Therefore, it is important to define a well entity-question template including entity and attribute information.

On the other hand, the above process is a pipeline, whose workflow consists of a single swim lane [5]. Therefore, we need to build templates for each step to describe the content and relationship between it and its adjacent steps.

2.3.1 Business goal template. For business goal, according to its depth and granularity, it can be defined at any level. But in practice, it is usually set intuitively and with a particular business target in mind. We have set 3 categories of high level business goals: enhancement of a positive target, reduction of a negative target, and achievement of official standard.

2.3.2 Entity-question list template. With the identified business goal, we enumerate entity-questions to obtain the quantitative information to achieve the goal. Before the questions can be set up, we should define the perspective and then, we list all entities addressed in four categories: inputs and resources, products and by-products, internal artifacts, and activities and flow paths [1]. All the questions are asked for the better understanding of an attribute that is related to the entity included in the questions.

2.3.3 Subgoal template. That grouping questions based on entities, and translating them into manageable subgoals is a convergent process. We define a set of grouping keywords and the corresponding subgoals to organize the questions.

2.3.4 Entity-attribute template. The elements of entity and attribute are derived from entity-questions. Therefore, the entity-attribute pairs are related to questions, subgoals, and grouping keywords. All the above tuples are included in this template.

2.3.5 Measurement goal template. We organize the five dimensions and formulize a measurement goal using the following template: Analyze the *ENTIEY* with respect to the *ATTRIBUTE* from the *PERSPECTIVE* for the *PURPOSE* in the *ENVIRONMENT*. [4]

2.4 Setting relational tables

For describing all elements and the relation among them in the five steps, we assume there is a mental big map. In this map, we would see that an identified business goal is supported by a set of questions, which include perspectives, entities, attributes, and grouping keywords. Indexed by grouping keywords, these questions are cataloged into some subgoals. With the prioritized and selected questions, three dimensions of measurement goals can be abstracted, and then with the purpose and environment information, formal measurement goals are formulized.

According to the pipeline style of goal-driven process, we can break the mental big map into some small tables. After the templates for the five steps are defined, we enumerate all meta-elements following the templates, and identify both input and output for each step. Further, we set some relational tables (Table 2. to Table 5.) to associate the isolated information between the adjacent steps to each other. In the ISMS project, roughly, we defined about 300 entity-questions, 50 subgoals, 300 pairs of entity-attribute, and 9000+ measurement goals to support 15 basic business goals.

Table 2. Business goal-question table

Business Goal	Questions		
Reduce cost	Q1. How many people are necessary?	...	Qn. … … ….

Table 3. Question – keyword - P/E/A table

Question	Grouping Keyword	Perspective	Entity	Attribute
How many people are necessary?	People	Project manager	People	Number

Table 4. Question - keyword - subgoal table

Question	Grouping Keyword	Subgoal
How many people are necessary?	People	Improve human resource management.

Table 5. Measurement goal table

Environment	Purpose	Perspective	Entity	Attribute
...	Predict	Project manager	People	Number

3. Knowledge processing

3.1 Goal question relationships

So far we have reverse engineered the process and extracted the knowledge and the possible solution for specific problems. Now it is necessary to apply the knowledge in a forward engineering way to help user with a business goal. The entity question table set will help us with this. Table 6 is a portion of entity-question weights form, which is a 3-D relational table. This table reflects the tightness degree of the relationship between the entity-questions and a specific business goal from different perspectives. In this table, all questions are given weights, whose scale is from 0 to 4; the larger the number means the tighter the relation. By the weight, the relation between questions and business goals is described and stored. For an identified business goal, with predefined criteria, a subset of related questions can be selected from the complete entity-question list.

3.2 System learning and updating

The Users of ISMS use the GQM analysis to receive a measurement plan from goals they entered into the system. They will receive feedback for each step of the GQM method. Particularly, there is also the option of skipping GQM steps if the result has been previously used. ISMS can record all the original business goals, and the information asked by system, also the solutions would be stored. Further, if the input information were repeated in any project, system would provide the final solution automatically based on the knowledge that system learned from previous projects. In this case, the human-machine communication in some GQM steps can be skipped.

In addition to general users' functionality, the expert users will be able to add, modify, and delete their own entries in their own changeable knowledge base. After reviewing and verifying by the administrator of ISMS, the knowledge will be combined with system knowledge base for update.

Table 6. Entity-question weights

Business Goal: **Enhance product quality**	**Perspectives**							
	User	Customer	Manager	Developer	Organization	Tester	Maintainer	Quality Manager
Questions								
Is personal turnover hampering our goal?	0	0	4	4	4	4	4	4
How many people are necessary?	0	0	4	4	4	3	3	3
Are the people over-worked?	0	0	4	2	4	2	2	4
How is productive of the people currently?	0	0	4	2	4	2	2	4
In what area do the people need improve?	0	0	4	2	4	2	2	4
How is efficiency of the people currently?	0	0	4	2	3	2	2	4
Are the people motivated?	0	0	4	2	3	2	2	4
How is experience of the people currently?	0	0	4	2	3	2	2	4
How much the people are paid?	0	0	4	2	4	2	2	3
How is performance of the people currently?	0	0	4	2	3	2	2	4

3.3 Combination Implementation

The heart of ISMS is the combination of the expert opinionated weights. These weights are used in determining the transition of the GQM process from one step to another. These weights were obtained through a survey where several expert was asked to determine the relevance of a question with a business goal and combined using Dempster-Shafer Theory of Evidence. To ensure accurate results, experts supply a confidence interval. This confidence interval is only supplied once for the total set of questions supplied by ISMS. The expert must then adhere to this level for information to be valid. See Table 7 for an example related-unrelated table.

Table 7. Experts' related / unrelated probabilities

Weight	**0**	**1**	**2**	**3**	**4**
Related	P_{A0}= 0.01	P_{A1}= 0.25	P_{A2}= 0.50	P_{A3}= 0.65	P_{A4}= 0.95
Unrelated	P_{A0}?= 0.98	P_{A1}?= 0.70	P_{A2}?= 0.40	P_{A3}?= 0.25	P_{A4}?= 0.05

To capture the GQM process in the perspective of the many different parties involved in the software development lifecycle, the GQM process can be performed from many different perspectives. In ISMS there are eight possible perspectives that can be chosen from user, customer, manager, developer, organization, tester, maintainer, and quality manager. The expert then needs to give weights using the scale above for each question for each perspective.

The combination process is defined as the following method:

If there are 2 experts, A and B, then the weight given by A is denoted as $W_A[i]$, $i \in \{0, 1, 2, 3, 4\}$. The possibility of Related of $W_A[i]$ is denoted as P_{Ai}, and the possibility of Unrelated is denoted as P_{Ai}'. Similarly, the weight given by B is denoted as $W_B[j]$, $j \in \{0, 1, 2, 3, 4\}$. The possibility of Related of $W_B[j]$ is denoted as P_{Bj}, and the possibility of Unrelated is denoted as P_{Bj}'.

The belief of Related after combining is expressed by [B_R B_R+Interval], and the belief of Unrelated after combining is expressed by the interval [B_{UR} B_{UR}+Interval].

$$B_R = \left[\frac{P_{Ai} \times P_{Bj} + P_{Ai} \times \left(1 - P_{Bj} - P_{Bj}'\right) + P_{Bj} \times \left(1 - P_{Ai} - P_{Ai}'\right)}{1 - \left(P_{Ai} \times P_{Bj}' + P_{Ai}' \times P_{Bj}\right)} \right] \quad (1)$$

$$B_{UR} = \left[\frac{P_{Ai}' \times P_{Bj}' + P_{Ai}' \times \left(1 - P_{Bj} - P_{Bj}'\right) + P_{Bj}' \times \left(1 - P_{Ai} - P_{Ai}'\right)}{1 - \left(P_{Ai} \times P_{Bj}' + P_{Ai}' \times P_{Bj}\right)} \right] \quad (2)$$

$$Interval = \left[\frac{\left(1 - P_{Ai} - P_{Ai}'\right) \times \left(1 - P_{Bj} - P_{Bj}'\right)}{1 - \left(P_{Ai} \times P_{Bj}' + P_{Ai}' \times P_{Bj}\right)} \right] \quad (3)$$

4. System Design and Implementation

Following the agent-based design methodology, the design of ISMS is needed to become more distributed and focus less on connection orientation. This means client and service agents should be able to connect to a centralized location and get forwarded to the destination they require. With this idea in mind an agent registry is added to the design to incorporate a centralized location for connection information. This allowed agents and clients to easily determine where the other parts of ISMS are. Methodologies of this type are sometimes referred to as peer-to-peer type applications except in this system the peers are not equal meaning they do not have the same abilities.

4.1 Agent Interaction Architecture

The interaction of the agents is extremely important for ISMS to function correctly as a whole. To look at the design in a little more detail there are seven service type agents, a registry and a client. Each can be seen in Figure 2.

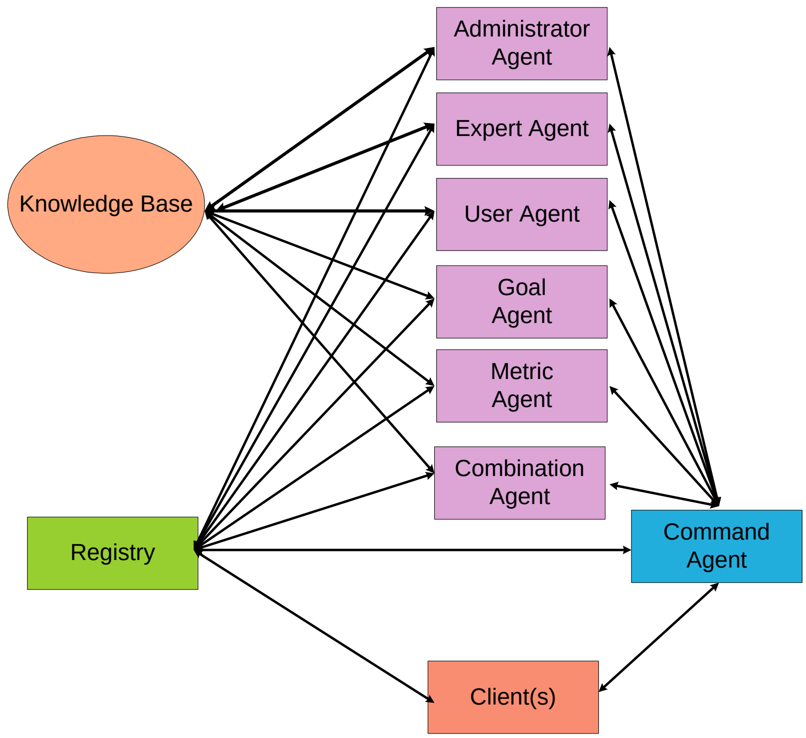

Fig. 2. ISMS High Level Agent Design

The registry will have a designated location using a hostname, which can be found using any DNS lookup. This allows the registry to appear to be stationary. Then the registry is ready to accept registrations. An agent then starts its services, and requests registration with the registry. When the agent gains registration, clients are then able to access the agents and receive service from the agents. Clients always query the registry to find a command agent, which is used to determine which agent to gain service from.

Each service type agent accepts requests and returns answers. The agents are:

- **Command Agent**: This agent is responsible for determining which services the calling agent needs. After it has decided which agent to use it calls on the registry to get the location of that agent, and then sends the message.

- **User Agent:** This agent is responsible for user accounts and user authentication. Other agents usually call upon the user agent to determine whether or not a user is valid or to alter or create user accounts. It has a direct reference to the database.
- **Goal Agent:** In ISMS the GQM process is split into two parts. The goal driven part and the metric driven part. This agent is responsible for the goal driven part. It queries the database and returns specific data needed usually by the client in the GQM process.
- **Combination Agent:** This agent is responsible for combining expert information to update the main ISMS knowledge base. The combination Agent is usually only called upon by an administrator.
- **Expert Agent:** Only expert users access the expert agent. This agent is responsible for updating experts personal knowledge bases and querying the current knowledge that is stored in an expert's knowledge base.
- **Administrator Agent:** The administrator agent is responsible for administrative tasks initiated by an administrator. These tasks include adding business goals, questions, sub-goals, entities and attributes to the knowledge base.
- **Metrics Agent:** This agent is responsible for the metrics driven part of the GQM process. This section of the GQM process includes step 6 to step 10. These steps were not required to be implemented in this project.

4.2 Advantages of the distributed agent design

With the current design there are many advantages in the user point of view as well as a performance and expandability point of view.

Taking the view of a user, using the registry with a distributed point of view allows the user to need to know less about the entire system to use it. This is because the registry acts as an abstraction point. This means that the user does not need to know where each of the different agents are at any time. Agents can go online and offline as they please and the registry will just forward the user to wherever the next command agent is. This is extremely advantageous from a distributed system point of view because the distributed architecture is completely dynamic and not fixed. The only problem that could arise is if there are no agents of a particular type online, this would cause the system not to function at all depending on the requested action.

Performance of web-based distributed system, performance is always an issue. So if certain agents in certain locations are becoming overwhelmed the registry can relocate users' requests to other agents of the same type. This allows for performance to be maximized.

Along with performance ISMS can easily be expanded because new versions of agents can be started and registered and the client can have absolutely no knowledge of this happening, because while a new version is being tested the clients can still use the old version of the agents. This allows for the agents to be expanded without the need for service interruptions.

4.3 Communication Medium

In ISMS the main communication medium is the Internet, but more specifically ISMS conforms to a specialized message-passing format XML (eXtensible Markup Language). This allowed the design of specific messages for ISMS.

XML is the heart of the communication of all of the parties in ISMS. To be able to communicate properly a message generator and a parser are needed.

4.3.1 XML generator

The XML generator is responsible for generating XML messages that will be able to be understood at the listening side. This requires a XML specification to be developed for the entire system. By doing such, messages can be understood and generated all throughout ISMS.

4.3.2 XML Parser

The XML Parser is used to decode the XML message into commands that ISMS agents understand. If the parser knows the XML specification then the agents can break the message into usable information and then in turn build a message using the XML generator in return.

4.4 Database Design

To store the data used in ISMS, a preexisting database management system was chosen. This allowed the use of mySQL, an already available database management system. The design of the database tables includes integrity constraints to ensure that data referenced between the database tables does not become corrupt or unusable. Since a preexisting database management system was used, interfacing software had to be written to query and receive information from the database. This was quite simple as a tool was already available to handle the task.

4.5 User Interface Design

When designing the graphical user interfaces (GUI) for ISMS intuition and simplicity were key design factors. The automation of the GQM process was to be displayed to the user in an intuitive fashion giving the user a straightforward approach to executing the GQM process.

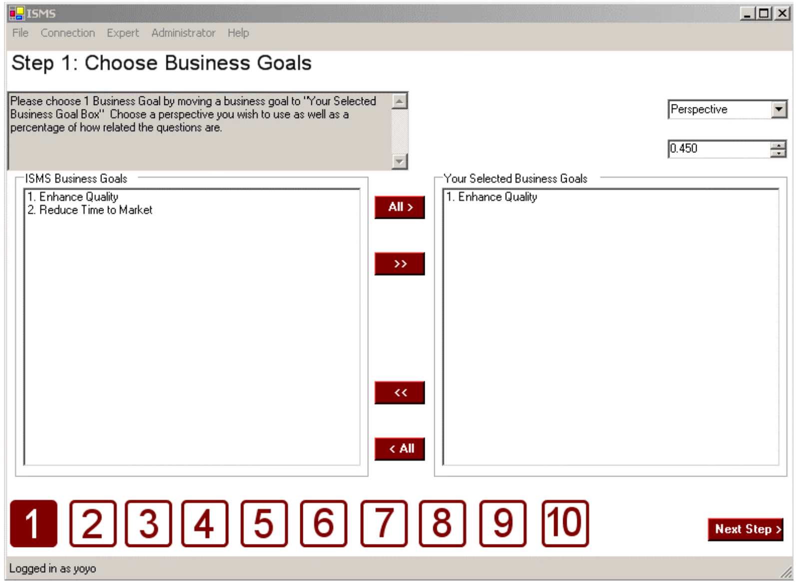

Fig. 3. ISMS Client User Interface

As can be seen in Figure 3, there is an instruction box that explains each step of the GQM process to the user, as well as simple buttons and images that make each GQM step easy to understand and execute.

To perform the GQM process, a user selects the items listed in left list box. When an item or items are selected, clicking on the '>>' button will move the items over to the select list box on the right. To continue through the current GQM step with the selected items on the right, the user must click on the 'Next Step' button.

To keep with the intuitive design, pre-existing Microsoft Windows interface design was followed to make the software easy to learn for users who are familiar with other Microsoft user interfaces.

5. Conclusions

The main purpose of ISMS is to automate the GQM process by building an intelligent system. This paper introduced a method to build knowledge base for this system. Via analyzing the software measurement domain, defining vocabulary, refining relationship, and setting relational tables, the knowledge about goal-driven process is represented. We processed the knowledge by weight tables, and further achieved system self-learning and updating functionality based on the knowledge base. The current agent-based design of ISMS could solve many of the problems with the previous designs and improved modifiability and performance of the ISMS system.

References

[1] R.E. Park, W.B. Goethert, and W.A. Florac, "*Goal Driven Software Measurement - A Guidebook,*" CMU/SEI, 1996.

[2] S. Russell, and P. Norvig, "*Artificial Intelligence: A Modern Approach,*" Pearson Education, 1995.

[3] L.C. Briand, S. Morasca, and V.R. Basili, *"An Operational Process for Goal-Driven Definition of Measures."* IEEE Transactions on Software Engineering, Vol. 28, No. 12, Dec. 2002.

[4] V.R. Basili, G. Caldiera, and H.D. Rombach, "*The Goal Question Metric Approach.*" in Encyclopedia of Software Engineering, Wiley, pp. 528-532, 1994.

[5] T. Chen, B.H. Far, and Y. Wang, "*Development of an Intelligent Agent-Based GQM Software Measurement System.*" 1st Int. Conf. on Agent-Based Technologies and Systems, Calgary, 2003.

[6] D. Zubrow, "*Measurement with a Focus: Goal-Driven Software Measurement.*" CROSSTALK: The journal of Defense Software Engineering, 1998.