International Journal of Hybrid Information Technology
Vol.9, No.2 (2016), pp. 147-162
http://dx.doi.org/10.14257/ijhit.2016.9.2.13

Analyze Requirements for PAIS with Best Practice

Juntao GAO, Xueshan Wang and Yongan WANG

School of Computer and Information Technology, Northeast Petroleum University, Daqing, 163318 China
gjt@nepu.edu.cn

Abstract

Traditional requirements analysis is focused on capture functions of the system that are being built from scratch. As the size and complexity of process-aware information system continues to grow, the use of COTS (Commercial off-the-shelf) software packages is being viewed as a solution. Use of COTS packages accelerates the implementation process to some extent, however, it puts enterprises into a dilemma: whether to adapt as-is business processes or to customize software packages. In order to help the enterprises get out of the dilemma, a methodology is proposed to automatically elicit software requirement using best practice as domain knowledge. The methodology involves three phases: business modeling, gap detecting and gap bridging. Firstly, VPML is employed to describe as-is business process, Secondly, semantics computing technology is employed to analyze the gap between enterprise needs and COTS software capacity. At last, Goal Reasoning Technology is employed to encourage enterprise users and software vendors take participate in decision making process together.

***Keywords**: Requirements Engineering, Best Practice, PAIS*

1. Introduction

PAIS (Process-Aware Information Systems) are those information systems that are executed on the basis of business process model, such as ERP, SCM, WfMS [1]. Due to the inherent complexity of PAIS, the implementation is always expensive and time-consuming [1]. Although procuring Commercial off-the-Shelf information systems is becoming increasingly popular in industry to save money and shorten time, the problem still exists. Research shows that even a best application package can meet only 70 percent of the organizational needs [2]. Some investigations have found that poor requirement elicitation is one of the main reasons for the failure [3]-[5]. The difficulty in eliciting requirements comes from both sides. First, enterprise users are not familiar with technologies of information and software. It is hard for them to propose right business requirements for the implementation of information system. On the other hand, it takes a long time for software vendors to grasp the specific needs of enterprise.

Traditionally, vendors took the software-driven approach to resolve the problem, such as SAP's ASAP [6], Oracle's AIM [7], [8] and SSA's One point [9], which are based on a common premise called "best practice". The "best practice" implies that software packages wrap the standard business process which has been proven the best in the practice. When gaps occur during implementation of PAIS, organizations are demanded to adapt itself to the software package rather than the other way around. It has been proven that the software-driven approaches do speed up the implementation, reduce the cost in some projects. However, research reports also indicate that the "best practice" premise is not always the case in practice. That is because every organization has some unique business processes which are the basis for their competitive edge. Daneva has analyzed the application practice of ASAP approach, and reported the problem [10].

ISSN: 1738-9968 IJHIT

Then the business-driven approach is proposed [11]. It emphasizes on the unique characteristic of business processes of the organization. The business-driven approach is more flexible than the solution-driven one. However it often leads to some unnecessary customizations for software packages, therefore it makes implementation more time-consuming and expansive. Apparently, it is an important issue to balance between customization and standardization in procuring requirements for PAIS. The standardization of business process reduces the development cost; however it cannot cover all aspects of business processes. On the other hand, the customization of software is more flexible. However it is too expensive and time-consuming.

In this study, a holistic methodology is proposed to elicit software requirements for PAIS implementation. The logic behind it is that business and software adapt for each other. That is, the user clearly represents business processes of their enterprise, and reference model are used to represent the business processes supported by PAIS. Then a meet-in-middle method is employed to elicit software requirements for PAIS.

The structure of this paper is organized as follows. In the next section, the framework of methodology is proposed to elicit requirements for PAIS with reference models. Section 3 gives a detail presentation of the method to make decision in requirements elicitation. Section 4 is the conclusion.

2. The Framework of Methodology

The main idea of this paper is that organizational needs and functionalities of software adapt for each other. As depicted in Figure 1, the method emphasizes on special needs of enterprise, so the business processes are clearly represented in VPML (Visual Process Modeling Language) which was initially developed by our laboratory in 1995 [12]. As-is business models just reflect the current state of businesses and the characteristics of the enterprise.

Reference models are knowledge accumulation of best practice and high level descriptions of capability of PAIS. The implementation is accelerated by reference models, but they are not always the right resolution of enterprise problems [3], [11], [13]. Therefore gap detecting becomes one of the key activities during requirement procurement.

Usually, as-is business models were built by enterprise users, while reference models were built by software vendors with different modeling conventions. Modeling conventions may differ in syntax and semantics. Difference in syntax is easy to overcome by using extant modeling transformation technology, such as Great, QVT-Partner and MT. However difference in semantics is more difficult to overcome. Enterprise users and Software vendors come from different domains and use different terminologies to build models.

In this study, semantic computing and pattern recognition are combined to resolve the problem. Conception distance based method is employed to measure the semantic similarity between basic elements, such as activities, objects and resources. Then pattern recognition is employed to measure the structure similarity between two models, which are composed of those basic elements.

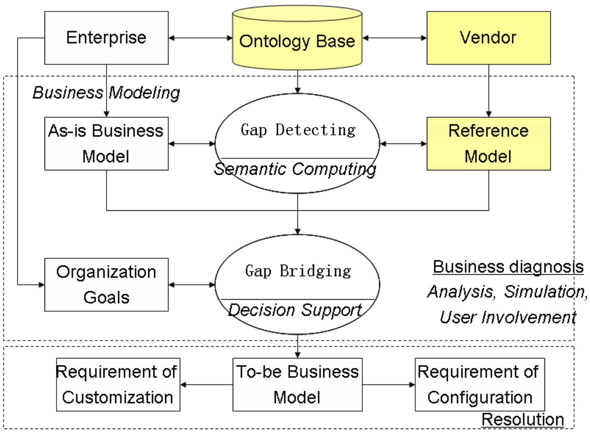

Figure 1. The Framework of Methodology

The result of gap detecting is a gap report, which depicts the gap between organizational needs and PAIS. The gap report also identifies the opportunity for business process changing. When business processes are not consistent with the " best practice" in reference models, it's difficult to decide whether to change business process or adapt software.

It is well known that implementation of PAIS is to satisfy various organizational goals more effectively and efficiently. Therefore, organizational goals are certainly the ultimate criterion to evaluate alternatives of business process. A formal goal model is used to declare the intent of organization. Different business processes have in general different impacts on the degree of satisfaction of higher-level goals, so a quantified reasoning technology with the goal model is proposed to determine the impacts. Because higher-level goals are usually more than one, the method of multiple attribute decision making is employed to determine to-be business processes.

3. Gap Detecting

3.1 Basic Idea

Various papers have studied the process similarity problem. In graph theory, graph isomorphism is often used to measure the similarity between two graphs [15]. However, the method usually only examine edges and nodes without catching the syntactical issues of business processes. The delta-algorithm is proposed to measure the difference between two models in database field [16]. Unfortunately, it still doesn't resolve the issues of the method mentioned above. In process algebra theory, Trace equivalence and bisimulation equivalence are usually employed to compare two process models [17]-[19]. A method based on trace equivalence is also proposed to assign weights to each trace based on execution logs which reflect the importance of a certain trace [20]. These methods determine whether two process models are identical or not, but they do not tell how much they differ.

Those methods mentioned so far don' t take semantic heterogeneity into account. Considerations of semantic similarities within business process models are reported only in fairly recent years. An ontology based method is proposed to compare business process models [21]. However the method does not take the structural difference into account that two process comprised by same activity set may also differ in structure.

In this paper, not only semantic similarity computation is employed, but also edit distance of process model is proposed to measure the gap of process models.

Although high level goals are used as ultimate criteria to evaluate business processes. In this paper, as-is business models of organization (EM for short) are constructed using a visual process modeling language VPML. VPML, described in detail in literature [22], has been applied in manufacturing industry for many years [23], [24]. VPML' s building blocks are three equally important classes of entities: activities, products and recourses. Activities and products are connected by dataflow link, while activities and recourses are connected by association link. VPML represents what activities are needed to accomplish a task, sequence of activities, input and output of activities, and who are responsible or participating for each activity in a graph. It is easy for users to describe their business processes in VPML. As an example, Figure 2 shows a process model of assembly warehousing of some aircraft factory.

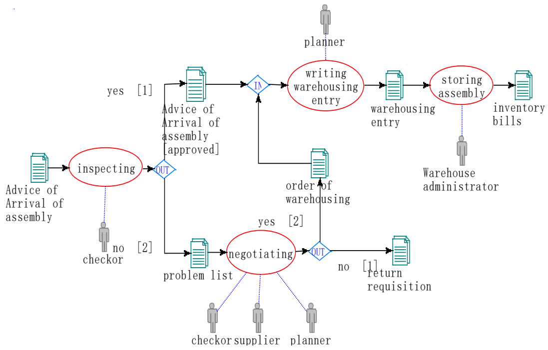

Figure 2. A VPML Model of an Assembly Warehousing Process

As mentioned above, the capability of PAIS is presented by reference model in business terms. Because PAIS is designed to serve a large variety of organizations rather than individuals, the reference model has many options supporting various business processes for different types of organizations. Rosemann designed a reference modeling language C-EPC based on classical EPC (Event-Driven Process Chains) [25].

Because a reference model is a kind of parameterized model involving more than one option, the gap between a reference model and a business model of organization equals to the minimum gap between all options and EM. Because this paper does not focus on the research of reference modeling language, so C-EPC is used as the reference modeling language here. After decomposing the C-EPC reference model, a simple mapping is made to transform EPC models into VPML models. Then the options in reference models are compared with EM to identify the minimum effort needed to adapt software to business processes.

In order to determine gaps between process models, the following steps are needed:

1) Picking out the mapping between corresponding elements of the models according to semantic similarity;
2) Measuring the gap of process models through the minimal effort to change model;
3) Generating the gap report according to the result of step 1 and step 2;

As a first step, such technologies of computing similarity as Edit Distance and WordNet are employed to pick out a mapping of business process models. The concept mapping is defined as a set of activity pairs (A1, A2), where A1 from one model and A2 from the other. Because of the effect of similarity spreading in process models, the idea of similarity propagation [26] is introduced to make the result of mapping more precise.

As a second step, the gaps between models are measured by counting the minimal necessary high-level change operations (*e.g.*, to add, delete or move activities). High-level change operations have richer syntactical meaning than primitive change operations (*e.g.*, to add, delete, move nodes or edges). This paper adopts the High-level change operations introduced by the ADEPT method [27] to guarantee syntactic correctness of models.

As a third step, gap reports are derived from the results of previous steps. The gap reports not only show the extent to which models differ, but also detail differences between their corresponding activities and data. According to the reports, users of organization and vendors make decision in fitting software to organizational needs more scientifically.

3.2 The Matching Algorithm

Business process models are a kind of graphs, and the semantic similarities between elements are greatly influenced by the relationship among them. Therefore, the idea of similarity propagation is introduced into matching algorithm of process models.

3.2.1 Formal Syntax of VPML

For the convenience of discussing the algorithm to compute similarity of models, a formal syntax of VPML is defined in this section firstly.

Definition 1: Given a business process model in VPML $P = (A_p, D_p, R_p, \Sigma_P, \Phi_P)$, A_p is the set of activities of P, D_p is the set of data of P, and R_p is the set of resources of P. $\Sigma_p \subseteq A_p \times D_p \cup D_p \times A_p$ is the set of dataflow of P, $\Phi_p \subseteq A_p \times R_p$ is the set of association links which connect activities and resources.

Definition 2: $Input : A_p \to 2^{D_p}$ is a function to define the relation between an activity and its input data. The constraint $SInput$ must be satisfied:

$$SInput \equiv_{def} \forall d \in Input(a) \cdot (d, a) \in \Sigma_p$$

Definition 3: $Output : A_p \to 2^{D_p}$ is a function to define the relation between an activity and its output data. The constraint $SOutput$ must be satisfied:

$$SOutput \equiv_{def} \forall d \in Output(a) \cdot (a, d) \in \Sigma_p$$

Definition 4: $Source : D_p \to 2^{A_p}$ is a function to define the relation between data and the activities that yield it. The constraint $SSource$ must be satisfied:

$$SSource \equiv_{def} \forall a \in Source(d) \cdot (a, d) \in \Sigma_p$$

Definition 5: $Target : D_p \to 2^{A_p}$ is a function to define the relation between data and the activities that consume it. The constraint $STarget$ must be satisfied:

$$STarget \equiv_{def} \forall a \in Target(d) \cdot (d, a) \in \Sigma_p$$

Definition 6: $Need : A_p \to 2^{R_p}$ is a function to define the relation between activities and the resources that they need. The constraint $SNeed$ must be satisfied:

$$SNeed \equiv_{def} \forall r \in Need(a) \cdot (a, r) \in \Phi_p$$

Definition 7: $Assign : R_p \to 2^{A_p}$ is a function to define the relation between resources and the activities that they take part in. The constraint $SAssign$ must be satisfied:

$$SAssign \equiv_{def} \forall a \in Assign(r) \cdot (a, r) \in \Phi_p$$

3.2.2 The Iterative Algorithm

Let $SimA(a_{p1}, a_{p2}) \geq 0$ be the similarity measure of $a_{p1} \in A_{p1}$ and $a_{p2} \in A_{p2}$ defined as a total function over $A_{p1} \times A_{p2}$. The determination of similarity is based on an iterative computation $SimA$. Let $SimA_i(a_{p1}, a_{p2})$ denote the similarity between activities a_{p1} and a_{p2} after i^{th} iteration. $SimA_0(a_{p1}, a_{p2})$ denotes the initial similarity between activities

a_{p1} and a_{p2}, which is obtained e.g., using word similarity computation of activity labels. The initial value of similarity influences the iteration times of the algorithm. In this paper, edit distance [28] and WordNet [29] is combined to measure the initial similarities.

Formula 1: the similarity of activity pair(a_{p1}, a_{p2}) after kth iteration,

$$SimA_k(a_{p1}, a_{p2}) = \omega SimA_{k-1}(a_{p1}, a_{p2}) + \frac{\omega_{Input} \sum_{i=1}^{|Input(a_{p1})|} \sum_{j=1}^{|Input(a_{p2})|} SimD_{k-1}(Input_i(a_{p1}), Input_j(a_{p2}))}{|Input(a_{p1})|\,|Input(a_{p2})|}$$

$$+ \frac{\omega_{Output} \sum_{i=1}^{|Output(a_{p1})|} \sum_{j=1}^{|Output(a_{p2})|} SimD_{k-1}(Output_i(a_{p1}), Output_j(a_{p2}))}{|Output(a_{p1})|\,|Output(a_{p2})|} +$$

$$\frac{\omega_{Need} \sum_{i}^{|Need(a_{p1})|} \sum_{j}^{|Need(a_{p2})|} SimR_{k-1}(Need_i(a_{p1}), Need_j(a_{p2}))}{|Need(a_{p1})|\,|Need(a_{p2})|}$$

Formula 2: the similarity of data pair(d_{p1}, d_{p2}) after kth iteration,

$$SimD_k(d_{p1}, d_{p2}) = \phi SimD_{k-1}(d_{p1}, d_{p2}) +$$

$$\frac{\phi_{Source} \sum_{i=1}^{|Source(d_{p1})|} \sum_{j=1}^{|Source(d_{p2})|} SimA_{k-1}(Source_i(d_{p1}), Source_j(d_{p2}))}{|Source(d_{p1})|\,|Source(d_{p2})|} +$$

$$\frac{\phi_{Target} \sum_{i=1}^{|Target(d_{p1})|} \sum_{j=1}^{|Target(d_{p2})|} SimA_{k-1}(Target_i(d_{p1}), Output_j(d_{p2}))}{|Target(d_{p1})|\,|Target(d_{p2})|}$$

Formula 3: the similarity of resource pair(r_{p1}, r_{p2}) after kth iteration,

$$SimR_k(r_{p1}, r_{p2}) = \gamma SimR_{k-1}(r_{p1}, r_{p2}) +$$

$$\frac{\gamma_{Assign} \sum_{i=1}^{|Assign(r_{p1})|} \sum_{j=1}^{|Assign(r_{p2})|} SimA_{k-1}(Assign_i(r_{p1}), Assign_j(r_{p2}))}{|Assign(r_{p1})|\,|Assign(r_{p2})|}$$

In above three formulas, ω, ω_{Input}, ω_{Output}, ω_{Need}, ϕ, ϕ_{Source}, ϕ_{Target}, γ and γ_{Assign} are called propagation coefficients ranging from 0 to 1 . They can be computed in many different ways.

The computation is performed iteratively until the Euclidean length of the residual vector $\Delta(SimA_n, SimA_{n-1})$ becomes less than ε for some $n > 0$. If the computation does not converge, it is terminated after a certain number of iterations.

3.2.3 Selecting the Mapping

In the last Section, all activity pairs are assigned values to denote similarities. This section focuses on the issue how to pick out the mapping that maximizes the sum of similarities. A mapping is a subset of activity pairs (A1, A2), in which A1 is from one model and A2 is from the other. The combinatorial explosion of the number of mappings makes the issue difficult to resolve.

In this research, Hungarian algorithm is used to solve the problem. It is about matching things that belong to two separate sets in a way that maximizes (or minimizes) the sum of some quantities. These quantities are typically benefits or costs associated with each matching. Each object from one set can be matched with only one object from the other set.

The complexity of Hungarian algorithm is $o(n^3)$ and the result is the minimum value, while it is expected to get maximum similarity. So the Hungarian algorithm is adapted to solve our problems.

(1) Constructing similarity matrix. Computing the similarity of the activity Ai in model A and the activity Bj in model B. Then assigning the value to the element (i, j) in similarity matrix.

(2) Subtracting off the row min from each row.

(3) Subtracting off the column min from each column.

(4) Starting with the row or column with the least number of zeros, marks one certain zero element and redlines the row and the column where the marked zero element exists.

(5) Repeating step 4 until each zero element is marked or redlined. If the number of marked zero elements is min(m, n), match the activity of model A in the row in which the zero element exists to the activity of model B in the column in which the zero element exists. Otherwise, go to step 6.

(6) Mark all rows without marked zero with *, and then mark all zero elements in rows with *, and mark all zero elements in columns with mark *, until mark * can not be added.

(7) Redline all the rows and columns without *.

(8) Identify the least one among the elements uncovered by lines and denoted by xij

(9) Subtract xij from the rows marked with *, and subtract xij from the rows marked with *, return to step 4.

3.3. Measuring the Similarity of Models

3.3.1 The Edit Distance of Models

Inspired by the idea of Levenshtein Distance of string comparing, the Edit Distance of business process models is defined as the minimal effort to transform a process model P into another model Q. Usually four steps are needed to realize this transformation:

According to the result of mapping, replace all activities being present in model P by corresponding activities in model Q.

Delete all activities being presented in model P, but not in the result of mapping.

Move all activities replaced in step 1 to locations as reflected by model Q.

Insert all activities being present in model Q, but not in the result of mapping.

The high-level change operations used in above four steps distinguish our methods from traditional similarity measures like graph isomorphism or sub-graph isomorphism, which only consider basic change primitives like the insertion or deletion of single nodes and edges.

After the mapping is selected, the number of necessary operations of replacements, insertions and deletions are easy to determine and hardly to reduce. However, there may be more than one way to move activities. Therefore, the key issue is how to use of the move operations to minimize the effort to change the structure of a process model. A method based on digital logic proposed in [30] is employed to determine the minimal number of move operations.

According to the number of high-level change operations, the distance of business process models is computed using the following formula:

$$Distance(P,Q) = \alpha \sum_{i=1}^{Count\ of\ Replace} 1 - Sim(A_i) + \beta Count\ of\ Delete + \chi Count\ of\ Move$$
$$+\alpha Count\ of\ Insert$$

α, β, χ are ranged from 0 to 1, and $2\alpha + \beta + \chi = 1$. They define the proportion of effort needed to make a kind of change operation.

3.3.2 Experiments

Although there is no standard way to evaluate computational measures of model similarity, one reasonable way to judge can be agreement with human similarity ratings.

In the experiments, twenty subjects was chosen and given 20 model pairs. The subjects were all experienced clerks in an airplane manufacturing enterprise. The models were sent to the subjects in different order by email. According to the judgments, the subjects choose one of results: identical (1), very similar (0.8), similar (0.6), different (0.4), very different (0.2), and absolutely different (0).

The average similarity over the twenty subjects was compared with the computational similarity measurement. The average difference is 0.09, and the biggest difference is 0.23.

4. Gap Bridging

Organizational goals are desired states or abilities that an organization wishes to achieve. In general, successful organizations often set high level and low level goals for service development, improving quality, reducing errors, or becoming more customer-focused. Organizational goals serve as an internal source of motivation and commitment and provide a guide to action as well as a means of measuring performance.

It is not easy to judge the impacts of business processes on organizational goal because of their large gap in abstraction level. Therefore, a goal-process model is employed to decompose high level goals into low level goals recursively until business processes are identified to fulfill them to some extent. Then an automatic reasoning method is employed to identify impacts of business processes on high-level goals.

For organizational goals, the business processes certainly contribute to the degree of their satisfaction, but do not guarantee their satisfaction. This is because an organization is a complex and huge system which is full of uncertainties. For example, " Order Fill Rate must not be higher than 85%" . If organization demands quality checking before yield products, the Order Fill Rate are certainly raised. But quality checkers may make mistakes, and they can' t guarantee the satisfaction of the goal for each order. Various goal modeling and reasoning methods have been proposed, such as i*[14] and KAOS [31]. But they are good at qualified reasoning, and limited to precisely specify the uncertainty between goals and processes.

Based on evidence theory, a formal goal-process model is given to specify the degree of goal satisfaction. Most other stochastic approaches demand probability distributions known, however in reality it is not always easy for a decision maker to specify the probability distribution in an inexact environment, while the application of evidence theory does not need so extreme conditions.

Section 4.1 gives a brief introduction to evidence theory. Then the method of quantified reasoning is proposed to compute the degree of satisfaction of high-level goals. At last, the alternatives of business process are given to help decision makers select the optimal business process.

4.1 Evidence Theory

The Dempster-Shafer theory is a mathematical theory of evidence based on belief functions and plausible reasoning, which is used to combine separate pieces of information (evidence) to calculate the probability of an event [32].

Dempster-Shafer theory contains the key concepts like: Frame of discernment is a set of all atomic propositions. Power set is the set of all subsets of the frame of discernment. Basic belief assignment (BBA) is a function like $m: 2^X \to [0,1]$ which satisfies $m(\Phi) = 0$ and $\sum_{A \in 2^X} m(A) = 1$.

In order to combine two independent sets of mass function, Dempster's rule of combination is proposed as follows:

$$m_1(\varnothing) \oplus m_2(\varnothing) = 0 \qquad (1)$$

Where

$$K = \sum_{B \cap C = \varnothing} m_1(B) m_2(C)$$

Compared with other probability theories, evidence theory has its particular advantages:

1) Evidence theory can shrink the hypothesis set by accumulating evidence.
2) Evidence theory can distinguish the unknown problem and the uncertain problem.
3) Evidence theory dose not need the empirical probability or the conditional probability.

4.2 Goal-Process Modeling and Reasoning

Given a goal-process model

$$M = (G, \Pi, \Sigma, \Gamma, P)$$

G is the set of goals of organization. Given a business process, there is usually no enough prior information to precisely determine the degree of satisfaction of g∈G. So a new interval indicator [$PSat$, $OSat$] is defined to measure the degree of goal satisfaction.

$PSat : G \to [0,1]$ denotes the pessimistic assessment of the satisfaction, that is, the least belief of the degree of the goal satisfaction according to the current evidences that contribute to the goal in the model. The evidences come from other goals and business processes.

$OSat : G \to [0,1]$ denotes the optimistic assessment of the satisfaction, that is, the maximal belief of the degree of the goal satisfaction according to the current evidences that hinder the goal in the model.

$\Pi \subseteq G \times 2^G$ defines the AND-decomposing relationships between goals. If goal g is AND-decomposed into subgoals g1,g2,g3,⋯,gn , then the satisfaction of all subgoals are the necessary condition for the satisfaction of goal g. In this paper, we adopt the simplifying hypothesis that the satisfaction of g1,g2,g3,⋯,gn are independent.

$$PSat(g) = \prod_{i=1}^{n} PSat(g_i) \qquad OSat(g) = \prod_{i=1}^{n} OSat(g_i)$$

$\Sigma \subseteq G \times 2^G$ defines the OR-decomposing relationships between goals. If goal g is OR-decomposed into subgoals g1,g2,g3,⋯,gn ,then at least one of the subgoals are satisfied so is goal g. Here, we also adopt the simplifying hypothesis that the satisfaction of g1, g2, g3,⋯, gn are independent.

$$PSat(g) = \max(PSat(g_1), PSat(g_2), \ldots, PSat(g_n))$$

$$OSat(g) = \max(OSat(g_1), OSat(g_2), \ldots, OSat(g_n))$$

P is the set of business processes supporting various goals. If a business process $p \in P$ is adopted, then PSat(p)=OSat(p) =1.

$\Gamma \subseteq G \times G \cup P \times G$ defines the impact between goals and business processes. If $(g_1, g_2) \in \Gamma$, then the satisfaction of g1 has impact on the satisfaction of g2. Γ^+ denotes the positive impact. If $(g_1, g_2) \in \Gamma^+$ then the satisfaction of g´ contributes to the satisfaction of g to some extent, but does not guarantee its satisfaction; Γ^- denotes the negative impact. $\Gamma^+ \cup \Gamma^- = \Gamma$ and $\Gamma^+ \cap \Gamma^- = \varnothing$.

$Weight : \Gamma^+ \to [0,1]$ denotes the weight of positive impact between goals.

$$Weight(g_1, g_2) = \omega$$

$$PSat(g_2) = PSat(g_1) \times \omega$$

$$OSat(g_2) = 1 - \omega + OSat(g_1) \times \omega$$

$Weight : \Gamma^- \to [0,1]$ denotes the weight of negative impact between goals.

$$Weight(g_1, g_2) = \varphi$$

$$PSat(g_2) = \varphi - OSat(g_1) \times \varphi$$

$$OSat(g_2) = 1 - PSat(g_1) \times \varphi$$

4.3 Combination of Multiple Impacts

As the matter of fact, impact relationships are a kind of main relationships widely existing goal-process models. Especially, there is more than one goal with impact on a goal, the combination of impact is not apparent to determine. Here, Dempster's rule of combination from evidence theory is employed to address the problem.

If $(g_1, g) \in \Gamma^+ \wedge (g_2, g) \in \Gamma^+$, then the framework of discernment is {S, D}, and S denote the satisfaction of g, and D denote the denial of g. Each impact on g is an evidence to judge whether it is satisfied or denied.

If $(g_i, g) \in \Gamma^+$ and $\omega_i = Weight(g_i)$ then

$$m_i(\{S\}) = PSat(g_i)\omega_i$$

$$m_i(\{D\}) = \omega_i - OSat(g_i)\omega_i$$

$$m_i(\{S, D\}) = 1 - m_i(\{S\}) - m_i(\{D\})$$

$$= 1 - \omega_i + (OSat(g_i) - PSat(g_i))\omega_i$$

else if $(g_i, g) \in \Gamma^-$ and $\omega_i = Weight(g_i)$ then

$$m_i(\{D\}) = PSat(g_i)\omega_i$$

$$m_i(\{S\}) = \omega_i - OSat(g_i)\omega_i$$

$$m_i(\{S, D\}) = 1 - m_i(\{S\}) - m_i(\{D\})$$

$$= 1 - \omega_i + (OSat(g_i) - PSat(g_i))\omega_i$$

According to the formula (2), the combinational impact of g1 and g2 is computed with m1({S})⊕m2({S}) and m1({D})⊕m2({D})

m1({S})⊕m2({S})=

$$\frac{m_1(\{S\})m_2(\{S\}) + m_1(\{S\})m_2(\{S,D\}) + m_2(\{S\})m_1(\{S,D\})}{1-(m_1(\{S\})m_2\{D\} + m_2(\{S)m_1\{D\})}$$ m1({D})⊕m2({D})=

$$\frac{m_1(\{D\})m_2(\{D\}) + m_1(\{D\})m_2(\{S,D\}) + m_2(\{D\})m_1(\{S,D\})}{1-(m_1(\{S\})m_2\{D\} + m_2(\{S\})m_1\{D\})}$$ PSat(g)= m1({S}) ⊕ m2({S})

OSat(g)=1- m1({D})⊕m2({D}) (3)

If there are more than two impacts on goal g, then

PSat(g)= m1({S})⊕m2({S})⊕⋯⊕mn({S})

OSat(g)=1- m1({D})⊕m2({D})⊕⋯⊕mn({D}) (4)

4.4 Decision Making

In this study, an interval is employed to describe the degree of satisfaction of organizational goals, however it brings new challenge to decision making. If two intervals are overlapped, it is not easy for decision makers to rank goals [33], [34]. For example, there are two business process cause different degree of satisfaction of a goal g. One is [PSat1(g),OSat1(g)], the other is [PSat2(g),OSat2(g)]. If PSat2(g)>OSat1(g), it is obvious that the second process is better than the first. However, if PSat2(g)<OSat1(g)<OSat2(g), it is difficult to draw conclusions. Paper [35] tries to convert an interval into a numeric value according to the attitude of decision makers. As a matter of fact, the attitude is hard to capture and quantify, therefore a new method is discussed below to rank without attitude factor.

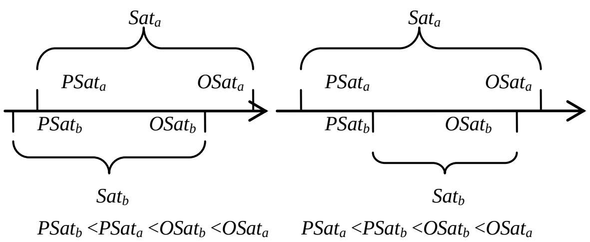

Figure 3. Two Cases of Overlapped Intervals

As Figure 3 illustrated, two cases of overlapped intervals are considered. In the left case, the two intervals cross one another; in the right case, Sata covers Satb. Both cases make it difficult to compare two intervals. Of course, another case of overlapped intervals still exists. That is, Satb covers Sata. The algorithm to compare them is the same as the right, therefore only the first two cases are analyzed next.

According to evidence theory, the length of interval decrease with evidences increase. Therefore the uncertainty of satisfaction of organizational goals could be decreased by collecting evidences. However the uncertainty will never be eliminated completely.

In order to compare two intervals, Sat is defined as an uniformly distributed random variable between PSat and OSat in this paper:

$$f(Sat) = \begin{cases} \dfrac{1}{OSat - PSat} & PSat \le Sat \le OSat \\ 0 & others \end{cases}$$

And $P_{Sat_a > Sat_b}$ denotes the possibility that Process Pa make more contributions than process Pb. Next, the algorithm to compute $P_{Sat_a > Sat_b}$ is given according to the cases of overlapped intervals.

When PSatb <PSata <OSatb <OSata:

$$P_{Sat_a > Sat_b} = \int_{OSat_b}^{OSat_a} f_a(x)dx + \int_{PSat_a}^{OSat_b} f_a(x)dx \int_{PSat_b}^{PSat_a} f_b(y)dy$$

$$+ \int_{PSat_a}^{OSat_b} f_a(x) \int_{PSat_a}^{x} f_b(y)dydx$$

$$\int_{OSat_b}^{OSat_a} f_a(x)dx = \frac{OSat_a - OSat_b}{OSat_a - PSat_a} \tag{5}$$

$$\int_{PSat_a}^{OSat_b} f_a(x)dx \int_{PSat_b}^{PSat_a} f_b(y)dy = \frac{OSat_b - PSat_a}{OSat_a - PSat_a} \times \frac{PSat_a - PSat_b}{OSat_b - PSat_b} \tag{6}$$

$$\int_{PSat_a}^{OSat_b} f_a(x) \int_{PSat_a}^{x} f_b(y)dxdy = \int_{PSat_a}^{OSat_b} \frac{x - PSat_a}{OSat_b - PSat_b}dx$$

$$= \frac{\frac{1}{2}(OSat_b^{\ 2} - PSat_a^{\ 2}) - PSat_a(OSat_b - PSat_a)}{OSat_b - PSat_b} \tag{7}$$

Therefore,

$$P_{Sat_a > Sat_b} = \frac{OSat_a - OSat_b}{OSat_a - PSat_a} + \frac{OSat_b - PSat_a}{OSat_a - PSat_a} \times \frac{PSat_a - PSat_b}{OSat_b - PSat_b} +$$

$$\frac{\frac{1}{2}(OSat_b^{\ 2} - PSat_a^{\ 2}) - PSat_a(OSat_b - PSat_a)}{OSat_b - PSat_b} \tag{8}$$

when PSata <PSatb <OSatb <OSata:

$$P_{Sat_a > Sat_b} = \int_{OSat_b}^{OSat_a} f_a(x)dx + \int_{PSat_b}^{OSat_b} f_a(x) \int_{PSat_b}^{x} f_b(y)dydx$$

$$\int_{PSat_b}^{OSat_b} f_a(x) \int_{PSat_b}^{x} f_b(y)dydx$$

$$= \frac{\frac{1}{2}(OSat_b^{\ 2} - PSat_b^{\ 2}) - PSat_b(OSat_b - PSat_b)}{(OSat_b - PSat_b) \times (OSat_a - PSat_a)} \tag{9}$$

$$= \frac{OSat_b - PSat_b}{2(OSat_a - PSat_a)}$$

Therefore,

$$P_{Sat_a > Sat_b} = \frac{OSat_a - OSat_b}{OSat_a - PSat_a} + \frac{OSat_b - PSat_b}{2(OSat_a - PSat_a)} \tag{10}$$

According to above formulas, the matrix of probability R is constructed.

$$R = \begin{pmatrix} \frac{1}{2} & P_{Sat_1 > Sat_2} & \cdots & P_{Sat_1 > Sat_m} \\ P_{Sat_2 > Sat_1} & \frac{1}{2} & \cdots & P_{Sat_2 > Sat_m} \\ \vdots & \vdots & & \vdots \\ P_{Sat_m > Sat_1} & P_{Sat_m > Sat_2} & & \frac{1}{2} \end{pmatrix} \quad (11)$$

If the scale of the matrix is not big, the ranking of satisfaction of goals may be done by human-beings. However, if the scale of the matrix is large, it is not easy to compare them yet. Formula (12) may be employed to compare them automatically [36].

$$S_i = \frac{\sum_{j=1}^{m} P_{Sat_i > Sat_j} + \frac{m}{2} - 1}{m(m-1)} \quad (12)$$

Although these formulas simplify the process of decision making, it also brings new risk into decision making. Obviously, the risk is inevitable because of lack of evidences. As is mentioned above, either positive impact or negative impact reduces the length of interval which reflects the degree of uncertainties. Therefore, the goal-process model should be checked before making the decision. That is, the organizational goal model start with the top-level goals, then sub-goals are identified recursively until business processes can be identified for fulfilling each of the sub-goals. At last, decision makers pick out their concerned high level goals. For each high level goals involved in the decision making, if $CSat(g) - PSat(g) > \sigma$, then the model need to be complemented, else the model is enough to make decision. $\sigma > 0$ is determined according to the problems of application.

5. Conclusions

In this paper a holistic framework of methodology to procure requirements for PAIS is proposed. Specifically, a quantified reasoning method with organizational goal-process model is proposed to support decision making. The work in this paper contributes not only to the ERP requirements engineering, but also to the application of goal reasoning in software engineering.

Although the methodology focuses on PAIS requirement procurement, it is also applicable to any other off-the-shelf requirement procurement. Now the method has been applied into an ERP implementation project. In the future, we propose to apply this methodology to more complex real cases to confirm its validity. We have developed a method to analyze the gap between enterprise needs and ERP systems automatically. The method still needs more experiment to verify. Next, a novel decision supporting method is being developed to make enterprise users and software vendors design business process together. The method is promising to improve the efficiency of requirements elicitation of ERP systems.

Acknowledgements

The research is supported by the Education Department of Heilongjiang province science and technology research projects (No. 1253G014).

The research is also supported by the Northeast Petroleum University youth science and technology research projects (No. 2013NQ118).

References

[1] C. H. Deutsch, “Software That Can Make a Grown Company Cry”, The New York Times, vol. 1, **(1998)**.
[2] M. Al-Mashari, “Process Orientation through Enterprise Resource Planning (ERP): A Review of Critical Issues”, Knowledge and Process Management, vol. 8, **(2001)**, pp. 175-185.
[3] C. Soh, S. S. Kien and J. Tay-yap, “Cultural Fits and Misfits: Is ERP a Universal Solution”, Communications of the ACM, vol. 43, **(2000)**, pp. 47-51.
[4] C. P. Holland and B. Light, “A Critical Success Factors Model for ERP implementation”, IEEE Software, vol. 16, **(1999)**, pp. 30-36.
[5] N. P. Dalal, M. Kamath, W. J. Kolarik and E. Sivaraman, “Toward an Integrated Framework for Modeling Enterprise Processes”, Communications of the ACM, vol. 47, **(2004)**, pp. 83-87.
[6] J. A. Hernandez, J. R. Elechiguerra, E. R. Bueno and S. A. Servera, “SAP R/3 Implementation Guide”, **(1999)**.
[7] ORACLE, “AIM AdvantageTM:A Comprehensive Method and Toolkit for Implementing Oracle's Packaged Applications”, ORACLE, **(1999)**.
[8] ORACLE, “Using Oracle TutorTM with AIM 3.0 and the Oracle Business Models”, **(1999)**.
[9] S. S. Global, “SSA OnePointTM Services: implementation methodology”, **(2006)**.
[10] M. Daneva, “Using Maturity Assessments to Understand the ERP Requirements Engineering Process”, presented at Proceedings of the IEEE Joint International Conference on Requirements Engineering, **(2002)**.
[11] C. Soh, S. K. Sia, W. F. Boh and M. Tang, “Misalignments in ERP Implementation: A Dialectic Perspective”, International Journal of Human-Computer Interaction, vol. 16, **(2003)**, pp. 81-100.
[12] Z. Bosheng and Z. Sheying, “Visual Process Modeling Language VPML”, Journal of Software (Supplement), **(1997)**, pp. 535-545.
[13] T. H. Davenport, “Mission Critical: Realizing the Promise of Enterprise Systems:”, HBS Press, **(2000)**.
[14] L. Chung, B. A. Nixon, E. Yu and J. Mylopoulos, “Non-functional Requirements in Software Engineering”, Kluwer Academic Publisher, **(2000)**.
[15] E. B. Krissinel and K. Henrick, “Common subgraph isomorphism detection by backtracking search”, Software: Practice and Experience, vol. 34, **(2004)**, pp. 591-607.
[16] W. M. P. van der Aalst, “Business Alignment: Using Process Mining as a Tool for Delta Analysis”, Requirements Engineering, vol. 10, **(2005)**, pp. 198-211.
[17] W. Yongxiang, W. Jinzhao and J. Jianmin, “Process algebra－symmetry and action decomposition”, Science Press, **(2007)**.
[18] J. Hidders, M. Dumas, W. M. P. van der Aalst, A. H. M. T. Hofstede and J. Verelst, “When Are Two Workflows the Same?," in Australaian Symposium on theory of Computing, ACM, vol. 41, **(2005)**, pp. 3-11.
[19] W. M. P. van der Aalst and T. Basten, “Inheritance of Workflows: An approach to tackling problems related to change”, Theoretical Computer Science, vol. 270, **(2002)**, pp. 125-203.
[20] W. M. P. van der Aalst, A. K. A. de Medeiros and A. J. M. M. Weijters, “Process Equivalence: Comparing Two Process Models Based on Observed Behavior”, presented at BPM'06, Vienna, Austria, **(2006)**.
[21] M. Ehrig, A. Koschmider and A. Oberweis, “Measuring Similarity between Semantic Business Process Models”, presented at the Fourth Asia-Pacific Conference on Conceptual Modelling, Australia, **(2007)**.
[22] Z. Bosheng and Z. Sheying, “Visual Process Modeling Language VPML”, Journal of Software (supplement) , no. 6, **(1997)**, pp. 535-545.
[23] P. Girard, Z. Li, C. Degang and D. Guy, “Concurrent Engineering and Enterprise Modelling Methodology - Application in Chinese Aircraft Companies”, the 16th World Computer Congress (WCC), **(2000)**.
[24] T. Wen-an, Z. Bo-sheng and Z. Li, "Research on the Flexible Simulation Technology for Enterprise Process Model”, Journal of Software, vol. 12, **(2001)**, pp. 1080-1087.
[25] M. Rosemann and A. W. Van, “A configurable reference modelling language”, Information Systems, vol. 32, **(2007)**, pp. 1-23.
[26] S. Melnik, H. Garcia-Molina and E. Rahm, “Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema matching”, presented at Proceedings of the 18th International Conference on Data Engineering, San Jose, CA, **(2002)**.
[27] M. Reichert and P. Dadam, “ADEPTflex-Supporting Dynamic Changes of Workflows Without Loosing Control”, Journal of Intelligent Information Systems, vol. 10, **(1998)**, pp. 93-129.
[28] P. Bouguet, M. Ehrig, J. Euzenat, E. Franconi, P. Hitzler, M. Krotzscn, L. Serafini, G. Stamou, Y. Sure and S. Tessaris, “Specification of A common framework for characterizing alignment”, Knowledge Web Consortium, **(2005)**.
[29] P. Pantel and D. Lin, “Discovering Word Senses from Text”, presented at Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining, **(2002)**.
[30] C. Li, M. Reichert and A. Wombacher, “On Measuring Process Model Similarity based on High-level Change Operations”, EWI-IS: Information Systems, **(2007)**.

[31] CEDITI, A KAOS Tutorial, **(2003)**.
[32] D. A. P, "Upper and lower probabilities induced by a multivalued mapping", The Annals of Statistics, vol. 28, **(1967)**, pp. 325-339.
[33] Z. Haijun, Z. Hongfu and L. Jian, "Application of Interval Multi-attribute Decision-Making Method to Aeroengine Performance Ranking", Journal of Southwest Jiaotong University, vol. 14, **(2006)**, pp. 182-186.
[34] A. Sengupta and T. K. Pal, "Theory and Methodology On comparing interval numbers", European Journal of Operational Research, vol. 127, **(2000)**, pp. 28-43.
[35] G. Juntao, Z. Li and W Zhiyao, "Decision Support for Handling Gaps between ERP functionalities and Organizational Needs: A Goal Reasoning Based approach", International Conference on Computer Science and Software Engineering. Wuhan, **(2008)**, pp. 255-258.
[36] D. Qingli and X. Zesui, "Singule-objective optimization model in uncertain multi-attribute decision-making", Journal of Systems Engineering, vol. 17, **(2002)**, pp. 50-55.

Authors

Juntao GAO, He received the PhD. degrees in Computer Science from BeiHang University in 2009. During 2009-2014, he stayed in Northeast Petroleum University to teach software engineering. His interest and research areas include process modeling, software requirements, semantic computing. Email: gjt@nepu.edu.cn.

Xueshan WANG, She is a graduate student to study software engineering in Computer Science from Northeast Petroleum University in 2014. Her interest and research areas include data quality, software requirements, process modeling. Email: 745785501@qq.com

Yongan WANG, master, He now works in the Northeast Petroleum University. His interest and research areas include information retrieval, clustering, and data integration. Email:wyajiessie9@126.com

.