RC24348 (C0709-007) September 14, 2007
Computer Science

IBM Research Report

A Comprehensive Semantic-Based Resource Allocation Framework for Workflow Management Systems

Chong Wang
Shanghai JiaoTong University
Shanghai, 200240
P.R.China

Zhile Zou, Qian Ma, Ru Fang, Hao Wang
IBM Research Division
China Research Laboratory
Building 19, Zhouguancun Software Park
8 Dongbeiwang West Road, Haidian District
Beijing, 100094
P.R.China

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

A Comprehensive Semantic-Based Resource Allocation Framework for Workflow Management Systems

Chong Wang
Shanghai JiaoTong University
Shanghai, 200240, P.R.China
wangchong@apex.sjtu.edu.cn

Zhile Zou, Qian Ma, Ru Fang, Hao Wang
IBM China Research Laboratory
Beijing, 100094, P.R.China
{zouzhile, maqian, fangru, wanghcrl}@cn.ibm.com,

***Abstract*— Resource allocation that targets to assign the appropriate resource to the dynamically generated work item has always been a critical issue in the workflow management system research. However, today's workflow systems almost still depend on the static literal-based matchmaking mechanism to support the resource allocation, which would suffer greatly from the diversity and mutability of resources in current open and dynamic network environment. In this paper, we propose a comprehensive semantic-based resource allocation framework to enhance the matchmaking process. The essential contributions of this framework are: firstly, an OWL ontology is provided to describe available resources based on their semantic information. Secondly, based on the ontology, effective semantic reasoning techniques are used to select the eligible resource candidates. Thirdly, a bidding approach is adopted to further optimize the resource selection according to runtime conditions such as availability, cost and etc., which would typically overcome the inflexibility of the static assignment policy in previous study. With these ontology modelling, semantic matchmaking and bidding mechanism, our framework provides better resource allocation functionality.**

I. Introduction

Workflow management systems [3], have been playing a significant role in helping modern corporations to become more and more competitive since the last decade. Such systems support the designing, enactment, and monitoring of computerized business processes, i.e. workflows. A workflow contains a group of various tasks which can be carried out in some pre-designed order. At runtime, the tasks pending to be performed for a specific workflow are usually called *work items* [1]. For a workflow management system, one of the core functions is to effectively utilize the organizational structure to allocate resources for those dynamically generated work items [1]. Existing systems usually adopt a group of heuristic rules for resource allocation, which, as emphasized in [2] [13], is rather limited and inefficient. However, from the academia side, "there is only a surprisingly small body of literature to deal with the problems in this area" [13]. In this paper we investigate the key challenges involved and propose our novel solutions as contributions to this area.

From our investigation, the major challenges for implementing effective resource allocation in workflow management systems are:

1) *Effective Resource Modelling*: In order to find suitable resources to perform work items, we need to correctly describe the types and properties of resources. Thus a resource model that properly organizes those types and properties is necessary. Nevertheless, in modern enterprises there are a great variety of resources that may participate in workflows, and there are usually complex relationships among the resources. What's more, those types, properties and relationships of resources may change over time. Therefore, it has always been a challenge to effectively model resources.
2) *Quantified Resource Matchmaking*: A work item may pose complex requirements on the possible performers, while resources have complex properties to describe them. For the same property, the work item requirement and different resources may posses different values. Thus, it is important to quantify the difference between available resources in order to find those qualified ones that fulfil the requirements. However, how to support such quantification is not trivial.
3) *Dynamic Information Capturing*: In modern enterprises, the resources are usually located distributively and their status are highly dynamic. It is impossible to provide a centralized database that captures and holds all real-time information about the resources. However, some dynamic information is vital in making an optimized allocation determination, e.g. the expected time and cost for a resource to perform the pending work item. Thus, it poses difficulties to properly capture such dynamic information for resource allocation.

In this paper, we propose a novel framework to systematically tackle the above challenges. Firstly, we use Semantic Web technologies [22] for resource modelling and management. Specifically, OWL (Web Ontology Language) [23] is adopted to define the concepts and properties about resources. A repository is devised to maintain the ontologies populated with enterprise resources. We also integrate an event monitory facility to dynamically update the modifications of resource information. Secondly, since we have ontology to model resources, we can to carry out semantic matchmaking rather than the rigid literal-based matchmaking. By properly defining the similarity of different concepts in an ontology, we conveniently quantify the extent that a resource fulfils the requirement of a work item. Thirdly, we integrate a bidding mechanism to obtain the previously unavailable dynamic information about resources. In such a mechanism, candidate resources provide their information to compete for a work item. A probabilistic model is provided at the end to synthesis both the statically retrievable information and the bidding information to make a more reasonable allocation determination.

Our contributions in this paper, thus, are: 1) We apply Semantic Web technologies to model and manage resources for resource allocation. Such technologies are based on sound logic theories and, as a modern candidate for knowledge management, have been successfully applied in many areas [18]. 2) We show how to carry out semantic matchmaking to help finding the qualified resources that fulfil the requirements of a pending work item. This support quantified resource matchmaking and effectively utilize the ontology resource model. 3) We integrate a biding mechanism to further optimize the allocation. It enables capturing previously unavailable dynamic information of resources.

The rest of the paper is organized as follows. Section II discusses the related works. Section III presents our framework, including the resource model, the semantic matchmaking methods and the bidding mechanism. In section IV, we show the whole resource allocation process through a case. At the end, section V concludes this paper and discusses the future work.

II. Related Work

Although resource allocation plays a central part in workflow management systems, there has been few literatures in this area. [4] is one among the earliest to concentrate on this problem. The authors presents the *Policy Resolution* framework for defining resource allocation constraints as policies, as well as how to resolve such rules at runtime. [5] investigates the requirements for flexible work allocation with the aim of supporting more control from the user side. However, there are no detailed solutions provided to fulfil such requirements. [8] presents a model to quantify and rank the suitability of a set of workers for a work item, based on three criteria: capabilities, social relationships and task relationships. [11] investigates the first step of training a task assignment classifier from workflow event logs, to reduce the burden of manual resource allocator. These works mainly focus on one aspect of the resource allocation problem and none of them pay much efforts on the resource modelling aspect. Different from that, we are aiming at providing a high level framework and thus we also pay a lot efforts to adopt modern Semantic Web technologies for resource modeling.

Most workflow systems assume that a work item is allocated to only one resource. However, in reality, many steps in a business process need multiple resources' participation at the same time. [6] is one of the few papers investigating such a problem. Another interesting problem in resource allocation arises when there are no resource available to allocate a work item to, and a preliminary solution can be found in [7]. Recently, M. Pesic et al. analysed the work distribution mechanisms of some popular workflow management systems using Colored Petri Nets [10] and proposed a reference model [9] as guidelines for implementing such a system. As claimed by the authors, they do not provide more advanced approaches, but only clarify the existing approaches to help understanding.

Applying market-based mechanism for resource allocation is not new. As early as the mid-nineties, [14] first proposed to adopt market-based approach for business workflow systems. A Gepperts et al [15] [17] presents a model to select the most suitable resource according to a balance on bidding cost and price. Our winner determination model is inspired by this, but we take more system information such as workload into consideration. What's more, our work differs from the previous in that, we consider the overhead of market-based approach and thus provide a novel hybrid strategy.

Besides, the resource allocation accuracy and efficiency can largely be affected by the modelling of the resources. [20] proposes a framework ARDE for the definition and modelling of workflow participants, i.e. the resources available for the workflow systems. ARDE contains a basic organization model to describe the resources and utilizes SQL statements for resource selections. However, we are encouraged to flexibly model the resources with Semantic Web technologies and performs the resource allocation based on the matchmaking of the resource descriptions.

III. Approach

A. Framework Overview

Fig. 1 depicts our resource allocation framework. The input for the system is a work item pool. It stores the generated work items by the workflow enactment engine that manages the running workflows. The *work item selector* is a component that follows the predefined policies to select the next work item to carry out resource allocation. The policies employed here may be FIFO (First-In, First-Out), EDD (Earliest Due Date), or the like. We allow the workflow system administrator to choose certain policies according to the actual situations. More detailed descriptions on using such policies are out of the scope of this paper and interested readers are referred to [2].

After a pending work item is selected, the *resource allocator* employees a bidding mechanism to allocate resource for it. As a first step, the pending work item is sent to the *candidate resource selector*. The selector broadcasts bidding requests to the available resources. Information about the work item and requirements on the candidate resources can be shipped with the bidding requests. The requirement information can be set by the process designer, or adjusted by the case manager when a new workflow is instantiated or even during runtime. Requirements shall be specified as mandatory or optional for the candidate resources to follow. After the bidding request is sent out, a timer is set up and the selector waits to collect the responses.

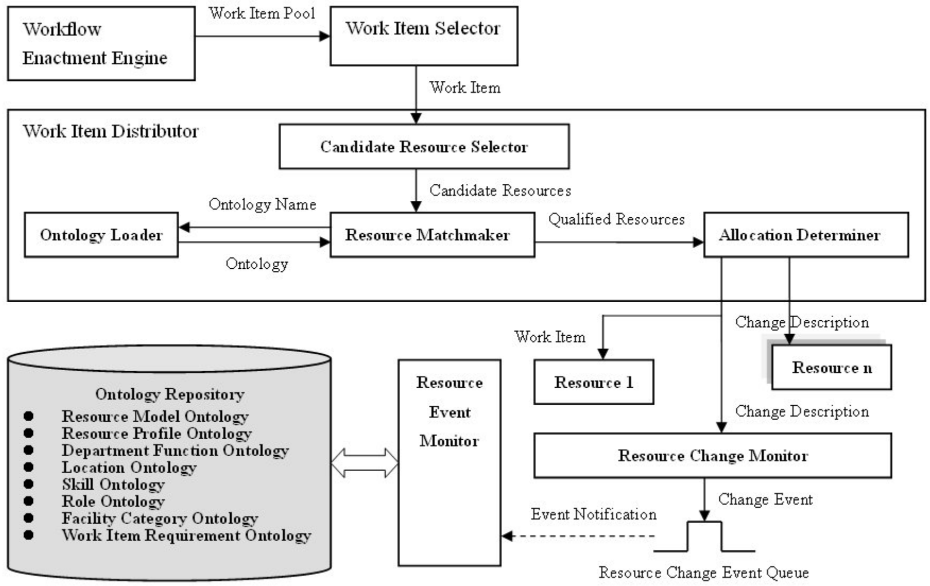

Fig. 1. The Architecture of the Framework

At the resource side, each candidate resource estimates its local status (capability, workload, interest, etc) and determines how to reply. In the replied message, each bidder is required to at least provide the time and cost it need to perform the work item. If for some reason, a resource is not able or do not want to process this work item, it can simply ignore the request. On the other extreme, even if the resource failed some requirements of the work item, it is still allowed to participate in the bidding, and even may win the bidding if it provides more attractive bidding cost and time. Correspondingly, in order to guarantee that an unqualified resource does not win by simply providing more attractive biddings, the *resource allocator* must to provide functionalities to verify all the replied resources, which will be detailed in subsection III-C.

When the predefined bidding time expires, the received responses are collected and the final determination on the winning resource will be made. In our framework, the winner determination is carried out in two major steps:

1) Firstly, the *resource matchmaker* is employed to identify the qualified resources and filter out those unqualified ones. This operation is based on semantic matchmaking techniques and mainly considers the statically retrievable information of the candidates. During the semantic matchmaking, the *ontology loader* is utilized to access the ontologies which are maintained in the *ontology repository* and stores all data about resource models.
2) Then, the *allocation determiner* synthesis the above semantic matchmaking result and the replied bidding information to determine the winning bidder through a probabilistic model.

At the end, the *allocation determiner* removes the pending work item from the work item pool and assigns it to the winning resource. The *allocation determiner* also notifies all the bidders the bidding result and sends an event to the public event queue to report the allocation event. As shown in the architecture, such an event is reported via the *resource change committer*, which, together with a public event queue, is assumed to be provided within the enterprise information infrastructure (e.g. IBM-MQ [24]). The *resource change committer* also accepts and reports the resource change events from other kind of information systems. Accordingly, aside with the resource ontology repository, an *event monitor* is also designed to monitor such resource change events and then dynamically updates the information for the resources.

In the following subsections, we will detail the major parts of our framework, namely the

B. The Resource Model

In our framework, OWL (Web Ontology Language) [23] is utilized to model various domain resources, such as Organization, Facility and Employee, etc. The relationships among the resources are modelled as properties of the corresponding concepts. Fig. 2 is generated with the Protégé Jambalaya [21] plugin, and shows a network of such modelled resources and relationships in our domain ontology. Each node in the figure stands for a concept in the ontology, while each arc represents the relationship between the corresponding concepts.

The concept Resource is an abstract concept that represents arbitrary resource in the domain, and the following essential concepts are derived from it:

1) Organization: This concept provides a means to model the structures of the organizations within a company. A tree structure of the organizations can be derived by modelling the parent-child relationships with the “belongsTo” property. Other organization attributes, such as location and functions, are also modelled within

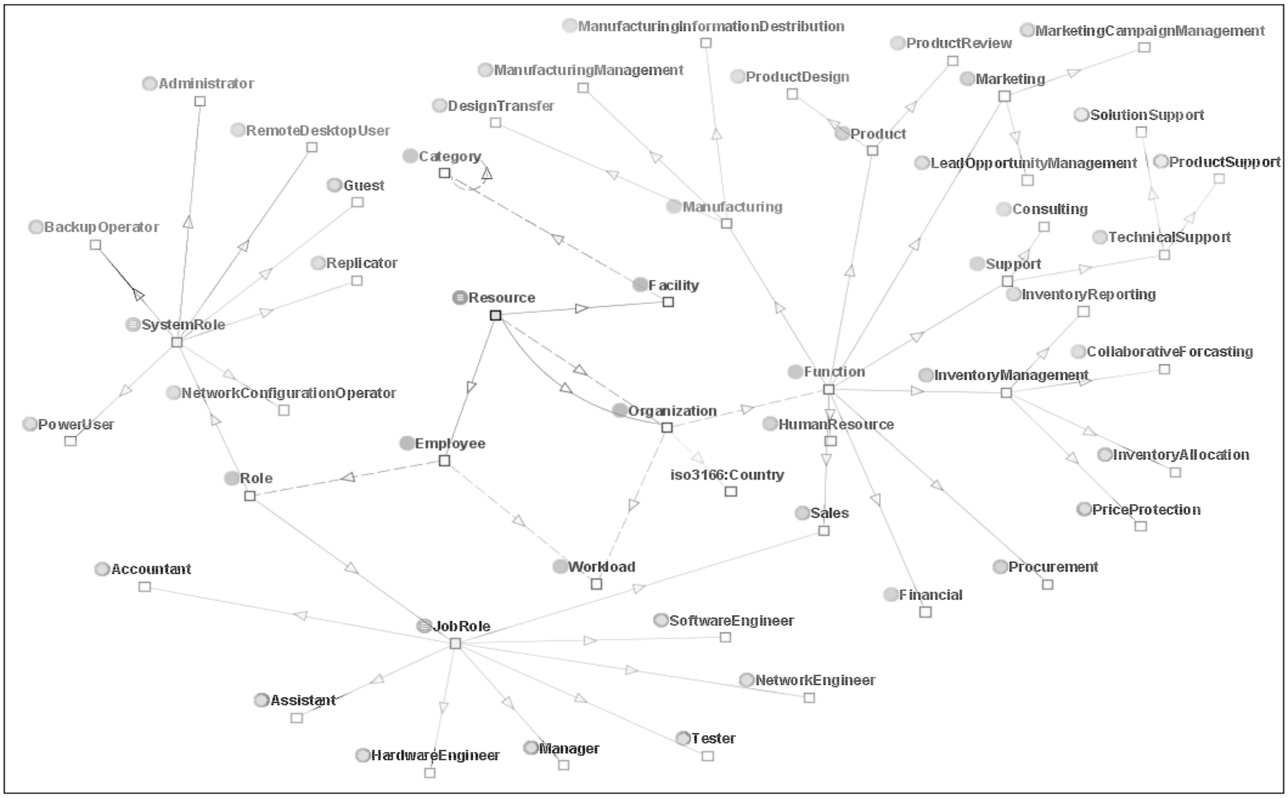

Fig. 2. The Domain Ontology Graph

the concept. The properties associated with the concept Organization are described in Table I:

TABLE I
THE PROPERTIES OF ORGANIZATION

Name	Description
departmentID	The identifier of the organization.
belongsTo	The identifier of the direct parent organization.
averageWorkload	The average workload of the entire organization.
location	A geography ontology entry that identifies the organization location
functions	The major jobs or technical areas of the organization. Each function is a function entry in a concrete function taxonomy ontology.

Organization is the largest resource unit in our resource model, which contains descriptions about the associated resources of smaller granularities, such as Employee and Facility. Thus, Organization is treated as the resource model, and instances of the concept Organization are regarded as resource profiles.

2) Employee: The concept Employee models each employee in different organizations with both static and dynamic metrics. The static metrics include skill set, role and name, etc; the dynamic metrics include workload and status that reflects the employee's runtime state. Table II shows more detailed descriptions over these metrics:

TABLE II
THE PROPERTIES OF EMPLOYEE

Name	Description
SN	The Serial Number of the employee.
name	The name of the employee.
belongsTo	The identifier of the organization the employee belongs to.
roles	The roles the employee plays in different context. Each role is a role entry in a concrete role taxonomy ontology.
status	Available/Unavailable
workload	The workload of the employee.
skills	The skill set of the employee. each skill is a skill entry in a concrete skill taxonomy ontology.

3) Facility: The concept Facility models all other resources except for employees within an organization. The facilities can be categorized to enable more efficient query and filtering operations. The properties of the concept Facility is shown in Table III.

In this way, the resource model not only describes the static aspects of the domain resources, but also reflects their dynamic characteristics that keeps changing at runtime. And given a specific work item, the selection and determination of the resources can be more precise, as introduced in the following subsection.

TABLE III
THE PROPERTIES OF FACILITY

Name	Description
SN	The Serial Number of the facility.
status	Available/Unavailable.
belongsTo	The identifier of the department that owns the facility.
category	An category entry in a facility category ontology.

C. Semantic Resource Matchmaking

Resource determination is the procedure that selects and verifies the resources that are qualified for executing the given work item. Current approaches like the ARDE framework [20] mainly focus on static characteristics of the resources and the determination technique is based on syntax value comparisons, e.g SQL based queries. However, we are encouraged to fully leverage the flexibility and matchmaking capability of OWL ontology, and incorporate the dynamic changing aspects of the resources in the resource determination procedure.

First, a Work Item Requirement model is introduced to describe the qualifications the expected resources must take to accomplish the work item successfully, the associated properties are shown in Table IV:

TABLE IV
THE PROPERTIES OF WORK ITEM REQUIREMENT MODEL

Name	Description
duration	The schedule constraint for the work item, specified with fromDate and toDate values.
locations	The locations of the organizations that are more suitable for processing the work item. (Optional)
maxAvrWL	The maximum averageWorkload of the organization
functions	The function the organization must have to complete the work item successfully. (Optional)
employees	For a department that has all the required functions, it must have all the expected employees with specific skills, available status and workload lighter than thresholds.
facilities	The facilities the department must own to complete the work item successfully. (Optional)

For the work item that requires team collaborations, the function constraints for each participant organization is necessary to be specified to guarantee effective collaborations; however, for the work item to be completed by a specific role person, the constraints are captured by the employee entries, i.e. the function constraints of organizations are no longer mandatory. Also, the facilities constraints are optional with the following considerations: 1) such constraints are not applicable for some cases like human approval; 2) the required facilities are too common in the organization and thus not necessary to be specially declared, e.g. the workstations within an IT company.

With the above work item requirement model, the determination is performed in two stages: Profile Matching Stage and State Evaluation Stage. The Profile Matching Stage performs matchmaking over the static characteristics of the resources, such as organization functions, employee skills and facility categories, and returns a set of qualified resources; the State Evaluation Stage further filters the qualified resources with their runtime state, including their status and workload. An essential operation in the characteristic matchmaking procedure is the calculation of the similarity between two ontology concept instances. Let c_1 and c_2 be two instances of the concepts C_1 and C_2 respectively, their similarity is calculated as follows when c_2 is matched against c_1:

$$SIM(c_1, c_2) = \begin{cases} \frac{1}{log_2^{d+2}} & C_1 \quad subsumes \quad C_2 \\ \\ \frac{1}{2^d} & C_2 \quad subsumes \quad C_1 \end{cases}$$

where d is the concept distance between C_1 and C_2. For two concepts that one subsumes the other, the concept distance is determined by the number of the hierarchical levels between them within the ontology hierarchy; if they are equivalent concepts, the concept distance d is 0; while the concept distance d is infinite if there is no subsumption relationship between them. Apparently, the similarity reduces dramatically fast when C_2 subsumes C_1 while the similarity reduces much slower on the opposite. This design consideration is out of the fact that a concept is refined by all the restrictions defined in its parent or ancestor concepts, but is not refined by the additional restrictions defined in its child concepts. Thus for three concepts C_1 subsumes C_2 and C_2 subsumes C_3, if the concept distance between C_1 and C_2 equals to the concept distance between C_2 and C_3, C_3 supersedes C_1 regarding the restrictions specified by C_2. As a consequence, two different similarity reduction rates are introduced as above.

Also, $SetSIM$ is introduced as a fundamental operation that calculates the similarity between two concept instance sets S_1 and S_2:

$$SetSIM(S_1, S_2) = [\Pi_{i=1}^{n} MaxSIM(S_1[i], S_2)]^{\frac{1}{n}}$$

where n=$|S_1|$ and m equals to the number of instances in S_1 that are matched with the instances in S_2. And $MaxSIM(c, S)$ is defined as follows, with c being a concept instance and S being a set of concept instances.

$$MaxSIM(c, S) = Max(SIM(c, S[j])), \\ for j = 1, 2, ..., |S|.$$

Then, let wr and org be two instances of the WorkItemRequirement and Organization concepts respectively, the function similarity between wr and org is calculated as follows:

$$SIM_{functions}(wr, org) = \\ SetSIM(wr.functions, org.functions)$$

Similarly, the employees and facilities similarities between wr and org are calculated as follows:

$$SIM_{employees}(wr, org) = \\ \frac{\sum_{i=1}^{|wr.employees|} EmplSIM(wr.employees[i], org.employees)}{|wr.employees|}$$

$$EmplSIM(empl, emplSet) = \\ Max(SetSIM(empl.skills, emplSet[i].skills))$$

$$SIM_{facilities}(wr, org) =$$

$$\frac{\sum_{i=1}^{|wr.facilities|} FaciSIM(wr.facilities[i], org.facilities)}{|wr.facilities|}$$

$$FaciSIM(faci, faciSet) = \frac{\sum_{i=1}^{|faciSet|} SIM(faci.category, faciSet[i].category)}{|faciSet|}$$

To this end, the overall similarity between the resource profile org and the work item requirement wr can be derived by consolidating the similarities returned by $SIM_{functions}, SIM_{employees}$ and $SIM_{facilities}$, as shown below:

$$ProfileSIM(wr, org) = (w_1 SIM_{functions}(wr, org) + w_2 SIM_{employees}(wr, org) + w_3 SIM_{facilities}(wr, org)) / (w_1 + w_2 + w_3)$$

where $0 \leq w_1, w_2, w_3 \leq 1$ are the weight values associated with $SIM_{functions}, SIM_{employees}$ and $SIM_{facilities}$, respectively.

In this way, the Profile Matching Stage is completed, and a set of qualified resources are ready for further optimization and filtering in the second stage, i.e. the State Evaluation Stage.

In the State Evaluation Stage, the states of the qualified resources are evaluated based on the given thresholds declared in the work item requirement wr, as shown below:

1) Organization: Let org be an Organization instance, then
   - $org.averageWorkload \leq wr.maxAvrWL$
   - $org.location$ is in one of the geographical areas specified by $wr.locations$.
2) Employee: Let $emplMatched$ be a skill matched Employee instance in the organization and $empl$ be the corresponding required employee described in wr, then
   - $emplMatched.status = Available$
   - $emplMatched.workload \leq empl.workload$
3) Facility: Let $faciMatched$ be a category matched Facility instance in the organization, then
   - $faciMatched.status = Available$

Finally, the ultimately qualified resources are sent to *Allocation Determiner* for further processing.

D. The Bidding Mechanism

In our framework, the resource allocation process is a bidding process. To start the process, the *candidate resource selector* sends the bidding requests and collect replies, which have been briefly described in subsection III-A. And in this subsection, we details how the allocation determination is made.

On receiving the qualified resources from the *resource matchmaker*, the *allocation determiner* takes a series of actions to select out the final target resource with the following considerations:

- The pending work item should be allocated with the most proper resource, which cost as least as possible and process the work item as well as possible.
- The resource allocation should contribute to pursue an optimal execution for the workflow (i.e. the case) and the whole workflow system.

By systematically considering the bidding information, the result of the semantic matchmaking as well as other dynamic information, the *allocation determiner* determines the final resource allocated for the work item. [15] proposed an algorithm to evaluate all the candidate bidders by considering the planned and actual time and cost status of the related workflow. [7] proposed a dynamic model to balance the quality and performance in resource allocation. In our framework, the determination is generated by following predefined policies. A refined algorithm based on that in [15] can be configured as the runtime determination policy. The rational in such a design is that there are different considerations for different types of work items, thus as a framework, we should allow defining different winner determination algorithms (policies). In the following we describes the available factors to consider when selecting the most proper resource. We will also propose an probabilistic model to synthesis all these factors.

- *Bidding cost and duration*: This is provided by the resources participating the bidding to "compete" for the work item.
- *Workload*: In our framework, the workload values for each resource is maintained dynamically. Intuitively, in order to balance workload among all resources, it is preferable to allocate a resource with relatively lower workload for a pending work item.
- *Suitability*: Suitability was first defined in [7], where it indicates the extent of a resource being suitable to process a work item. The authors assumed that some tables are maintained containing such data, which are usually manually set by experts. However, in reality, such data is hardly available due to the dynamic nature of enterprise resources. In our framework, we define the suitability as the matchmaking score, which dynamically quantifies the matching extent between descriptions of a resource and a work item requirement. The introduction of semantic matchmaking eliminates the need for a separately maintained table.

In addition to the policy based infrastructure, we propose to use a probabilistic model to determine the final winning resource. In our point of view, to choose a candidate resource as the winner can be treated as a conditional probabilistic event:"*given a work item requirement file W, a set of candidate resources S and the current workflow P, what is the probability to choose a candidate resource R*", which can be expressed as $P(R|W,S,P)$. We assume that the occurrence of S,P and W are independent events. Then by applying twice the Bayes's transformation, we obtain:

$$P(R|W,S,P) = \frac{P(R|W)P(R|S)P(R|P)}{P^2(R)}$$

We assume that the priory probability $P(R)$ is the same for each candidate resource R in the current resource set, thus we get:

$$P(R|W,S,P) \propto P(R|S)P(R|P)P(R|W)$$

Now we can find the proper R by maximizing:

$$P(R|W)P(R|S)P(R|P)$$

In our framework, $P(R|W)$ is interpreted as proportional to the similarity score computed out by the *resource matchmaker*. A resource that matches the requirements better is given a higher probability to choose. $P(R|P)$ is interpreted as proportional to the reciprocal of the workload score of each resource, and this way we favour a resource with smaller workload. $P(R|P)$ is interpreted as proportional to the value of the winning bidder determination function in [15], i.e. $C^{COR(wi)} \times D^{TOR(wi)}$, where C and D denotes the bidding cost and duration of the resource, $COR(wi)$ and $TOR(wi)$ means the *cost overrun ratio* and *time overrun ratio* of the current workflow and till the current work item (here we employ a set of different denotations in order to keep compatible with other parts of the paper; in the expression, C and $COR(wi)$ correspond to P and $COR(wf; A)$ in [15] respectively, while D and $TOR(wi)$ correspond to T and $TOR(wf; A)$ in [15] respectively. Please refer to [15] for more details about the definition and calculation). Therefore, the winning resource is the one that maximize:

$$ProfileSIM(W, R) \times (1/WL) \times (C^{COR(wi)} \times D^{TOR(wi)})$$

If there are multiple qualified resources ranked the first, we randomly select one as the final winner.

IV. Case Study

In this section, we elaborate the whole resource allocation process within our framework through a case.

The background of the case is a software development project, which follows a predefined process and is managed by a workflow management system. One task (at runtime, it will become a *work item* that triggers the resource allocation) in the process is to develop a web service, of which the functional requirements are detailed in a specification document and the non-functional requirements are specified in a work item requirement file, as listed in Table VI (As an example, we only list some major representative properties in the case, and the same holds for the data about the resources below).

TABLE VI

Work Item Requirements

Property Names	**Property Values**
duration	35
locations	Any
maxAvrWL	0.8
functions	Software Dev
employee1.skill	Web Service Dev
employee2.skill	Java Dev, Web Service Testing
facilities.category	Relational DB, Unix Compatible OS

The time when such a work item is produced at runtime, the *candidate resource selector* broadcasts the the requests to the available resources to solicit biddings. All the resources check the work item and 4 resources decide to participate in the bid and reply before the expiration time. The bidding costs and durations are listed in Table V. In the table, all the other information are retrieved from the populated resource model.

Then these data are sent to the *resource matchmaker* to identify the unqualified resources. In this example, Res#1 is filtered out because its bidding duration is greater than the upper limit specified in the work item requirement file. After that, the resource matchmaker calculates the similarities between those resources and the work item requirement. Table VII shows the intermediate results of the calculation. Take the SIM_{func} value of Res #3 as an example: the work item requirement file requires "Software Dev" function, while the resource has "SOA R&D" that is subsumed by "Software Dev" with a distance $d = 1$; thus, according to the semantic similarity function defined in section III-C, we get a score of 0.50. The other values are calculated in a similar way. Due to space limit, we omit those details here.

After the semantic matchmaking, the 3 left resources are sent to the *allocation determiner*. We assume that for the current workflow $COR = 1.2$ and $TOR = 0.7$, which means the aggregate cost is now larger than planned and the aggregate time is still well below budget. Thus we should consider relatively more about the cost and less on the time when allocating resource for the current work item. After the calculation with the available data, we obtain the final scores 0.036, 0.038 and 0.035 for Res#2, #3 and #4 respectively. Thus Res #3 is finally selected as the winning resource to perform the work item.

TABLE VII

Semantic Matchmaking Result

	SIM_{func}	SIM_{empl}	SIM_{faci}	$ProfileSIM$ ($w_1 = w_3 = 0.3$, $w_2 = 0.4$)
Res #2	1.00	0.89	0.50	0.81
Res #3	0.50	0.75	0.50	0.60
Res #4	1.00	0.71	0.00	0.58

By investigating the data, we can see that the allocation result is reasonable. Res #3 requires relatively less cost, has the lowest workload and relatively better fulfils the work item requirements. This elaborates the rationality of our allocation model.

V. Conclusion

This paper presents a novel semantic-based comprehensive framework for resource allocation in workflow management systems. By applying modern Semantic Web technologies, we effectively model enterprise resources for workflow resource allocation. With semantic matchmaking techniques, which in turn makes good use of the ontology resource model, we successfully quantify the difference between a set of candidate resources that helps select the qualified ones. The integration of the bidding mechanism is shown to be effective to capture more information about resources in a open and dynamic environment. In all, our framework tackles the major the challenges to implement effective resource allocation.

By integrating the bidding mechanism, we assume that the candidate resources have the intelligence to estimate their own status and provide reasonable biddings on cost and time. Thus, our allocation framework does not apply to arbitrary resource

TABLE V
INFORMATION OF THE BIDDING RESOURCES

	Department Functions	**Employees**		**Facilities Categories**	**WL**	**Bidding <Cost,Duration>**
		Name	**Skills**			
Res #1	Software Dev	Jimmy	Web Service Dev, J2EE Dev	IBM DB2, Sun Solaris	0.78	<30,40>
		Mark	Java Dev, Web Service Testing			
Res #2	Software Dev	Lily	Web Service Dev	Oracle, Red Hat	0.72	<55,15>
		Joe	J2SE Dev, Web Service Testing			
Res #3	SOA R&D	Tom	SOA Dev, J2EE Dev	IBM DB2, Ubuntu	0.55	<38,20>
		Jack	Java Dev, Web Service Testing			
		Ann	Web Service Testing, J2EE Dev			
Res #4	Software Dev	Mary	J2SE Dev, Web Service Testing	IBM DB2, Windows XP	0.68	<30,23>
		Kelvin	SOA Dev, Web Service Testing			

allocation in workflow management systems. In future work, we are considering to design a hybrid allocation approach that leverages the advantages of our approach and the existing approaches. We will investigate how to help specify or automatically determine whether to allocate resource for a work item in a bidding approach like ours or not. We will also quantitatively study the difference between our approach and existing ones through experiments.

REFERENCES

[1] Workflow Management Coalition: Workflow Management Coalition Terminology Glossary. Document No. WFMC-TC-1011. 3rd ed. Winchester 1999.

[2] W.M.P. van der Aalst and K.M. van Hee, Workflow Management: Models, Methods, and Systems, MIT Press: Cambridge, MA, 2002.

[3] D. Georgakopoulos, M. F. Hornick, and A. P. Sheth. An Overview of Workflow Management: from Process modelling to Workflow Automation Infrastructure. Distributed and Parallel Databases, 3(2):119-153, April 1995

[4] C. Bussler and S. Jablonski. Policy Resolution for Workflow Management Systems. In Proceedings of the 28th Hawaii International Conference on System Sciences, page 831. IEEE Computer Society, 1995.

[5] T. Herrmann, M. Hoffmann, Augmenting self-controlled work allocation in workflow-management-applications, in CSCW'98 Workshop on Adaptive Workflow Systems, Seattle, WA, Novermber 1998.

[6] W.M.P. van der Aalst and A. Kumar. A Reference Model for Team-Enabled Workflow Management Systems. Data and Knowledge Engineering, 38(3):335C363, 2001

[7] A. Kumar, W.M.P. van der Aalst, and H.M.W. Verbeek. Dynamic Work Distribution in Workflow Management Systems: How to Balance Quality and Performance? Journal of Management Information Systems, 18(3):157C193, 2002.

[8] Shen, M., Tzeng, G.-H., Liu, D.-R.: Multi-Criteria Task Assignment in Workflow Management Systems. In: Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS-36), Hawaii (2002) 202C210

[9] M. Pesic and W.M.P. van der Aalst. Towards a Reference Model for Work Distribution in Workflow Management Systems. In E. Kindler and M. Nuettgens, editors, First International Workshop on Business Process Reference Models (BPRM'05), pages 30-44, Nancy, France, September 2005.

[10] Pesic, M. and Aalst, W.M.P.v.d., modelling Work Distribution Mechanisms Using Colored Petri Nets. In Proceedings of the Sixth Workshop on the Practical Use of Coloured Petri Nets and CPN Tools (CPN 2005), (Aarhus, Denmark, 2005), 157-176.

[11] Liu Yingbo, Wang Jianmin, and Sun Jiaguang. A Machine Learning Approach to Semi-automating Workflow Staff Assignment. In SAC '07: Proceedings of the 2007 ACM symposium on Applied computing, pages 340-345, New York, NY, USA, 2007. ACM Press.

[12] Y. Chevaleyre, P. E. Dunne, U. Endriss, J. Lang, M. L. tre, N. Maudet, J. Padget, S. Phelps, J. A. R. guez Aguilar, and P. Sousa. Issues in multiagent resource allocation. Informatica, 30:3C31, 2006.

[13] Russell, N., van der Aalst, W.M.P., ter Hofstede, A.H.M. and Edmond, D., Workflow Resource Patterns: Identification, Representation and Tool Support. In Lecture Notes in Computer Science, (2005), 216-232.

[14] Harker P T and Ungar L H: A market-based approach to workflow automation, http://www.cis.upenn.edu/ ungar/NFS.html

[15] A. Geppert, M. Kradolfer, and D. Tombros. Market-based Workflow Management. International Journal of Cooperative Information Systems, World Scientific Publishing Co., NJ, USA, 1998.

[16] D. W. Judge, B. R. Odgers, J. W. Shepherdson, and Z. Cui. Agent-enhanced workflow. BT Technology Journal, 16(3):7985, 1998.

[17] Andreas Geppert. modelling Electronic Workflow Markets. In ER '00: Proceedings of the Workshops on Conceptual modelling Approaches for E-Business and The World Wide Web and Conceptual modelling, pages 5263, London, UK, 2000. Springer-Verlag.

[18] D. Fensel, C. Bussler, Y. Ding, V. Kartseva, M. Klein, M. Korotkiy, B. Omelayenko, R. Siebes, Semantic Web Application Areas, In: Proceedings of the 7th International Workshop on Applications of Natural Language to Information Systems, Stockholm - Sweden, June 27C28, 2002.

[19] Muehlen, M.Z. Organizational Management in Workflow Applications C Issues and Perspectives. Information Technology and Management 5(3-4). 271 - 291.

[20] Qijun Deng, Jiangtao Liu and Qing Zhang: ARDE: A Novel Framework to Define Workflow Participants. Proceedings of the 2005 IEEE International Conference on e-Business Engineering(ICEBE'05)

[21] Jambalaya: http://www.thechiselgroup.org/ chisel/projects/jambalaya/jambalaya.html

[22] Semantic Web: http://www.w3.org/2001/sw/

[23] W3C: OWL Ontology Language Reference, http://www.w3.org/TR/owl-ref/

[24] IBM-MQ: http://www-306.ibm.com/software/integration/wmq/

[25] SPARQL Query Language for RDF: http://www.w3.org/TR/rdf-sparql-query/