RI12008 07 March 2012 Computer Science

IBM Research Report

An Efficient Method to Formulate, Solve and Reuse Resource Allocation Problems Using Semantic Models

Pranav Gupta and Biplav Srivastava

IBM Research Division
IBM India Research Lab
4, Block C, I.S.I.D. Campus, Vasant Kunj
New Delhi 110070, India.

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

An Efficient Method to Formulate, Solve and Reuse Resource Allocation Problems Using Semantic Models

Pranav Gupta and **Biplav Srivastava**
IBM Research - India, New Delhi & Bangalore, India
{prguptan,sbiplav}@in.ibm.com

Abstract

Resource allocation is a common problem in industry and real world wherein the demand for resources is matched to supply while optionally optimizing some objectives. However, setting up such problems for efficient solving is time-consuming and error-prone because there is a diverse set of techniques that could be applicable depending on subtle problem variations. In this paper, we seek to tackle this by creating a semantic model of demand, supply and allocation aspects of the problem. Then using code-generation utilities for standard solvers and semantic queries, we show that one can create new allocation problems rapidly, reuse results for known allocation instances while setting up new problems, discover problem characteristics quickly and understand deep similarity among problems. The breadth of allocation problems we consider are job-shop scheduling, tackling forest fires, assigning people to IT tasks in a service delivery center and evacuating people and goods. Thus, using semantic technologies, we are able to extend the reach of allocation techniques to more real world applications.

Introduction

Much of the effort in solving a scheduling (and planning) problem is the time needed to set up such problems. In order to reduce it as well as make the whole process less error-prone, knowledge engineering for planning and scheduling has taken off in recent years with there now being even a competition(ICAPS-KE 2011) to compare such tools. We are especially interested in efficiently solving the *resource allocation* problem(Riley 1996; Beck and Fox 1998) which is a common class of problems seen in industry and real world.

In these problems, the *demand* for resources is matched to their availability or *supply* in order to achieve the most effective *allocation*. There exists a comprehensive summary of the past thirty years of research on algorithmic aspects of the allocation problem and its variant (Ibaraki and Katoh 1998). The breadth of allocation problems span job-shop scheduling(Graham 1966), tackling forest fires(Bratten 1970), assigning people to IT tasks in a service delivery center(Dixit *et al.* 2009), evacuating people(Inampudi and Ganz 2009; Wang *et al.* 2008) and excavating earth during road construction(Ji *et al.* 2010).

Some of the major issues in solving allocation problems are the following: (1) Setting up such problems for efficient solving is time-consuming and error-prone. This is because there is a diverse set of techniques that could be applicable depending on subtle problem variations. (2) There is little or no reuse of information sub-models which occur frequently. (3) Problem solving is highly dependent on availability of experts, although the solvers, that automatically solve on the models, are easily available. Example: LP(AMPL 2011), ILP(ILOG 2011), SAT(Moskewicz *et al.* 2001).

In this paper, we aim to capture common patterns from allocation problems, create a reusable semantic model to represent these patterns and reuse them while solving current and new problems types. The fact that semantic modeling can enable information sharing, especially on the web, has long been articulated (Berners-Lee *et al.* 2001). However, surprisingly, this approach has not been used for resource allocation. Using the models and and analysis techniques like semantic queries and model comparison, we can reuse and create problem instances faster. Furthermore, we have developed utilities to automatically create solver inputs from semantic models. Thus, we address the above issues and employ semantic technologies to extend the reach of allocation techniques to more real world applications.

Our contributions are that we:

- identify common information requirements in allocation problem
- develop an ontology called **R**esource **A**llocation **M**odel (**RAM**) to capture the demand, supply and allocation models
- demonstrate the generality and benefits of the created semantic models
- address issues of formulating and solving allocation problems using the created ontology with a decision support tool and methodology, respectively

In what follows, we first define the terminology used, motivate a sample of allocation problems: job-shop scheduling, assigning people to IT tasks in a service delivery center and tackling forest fires, and demon-

#	Term	Description	Examples
1	Problem	The analytical problem solved	Resource allocation
2	Problem domain	Real-world situation for problem	Work force matching, Fire-fighting
3	Problem class	Type of problem objective	Maximal matching, demand satisfiability, maximum utility
4	Scenario	Set of parameterized problem instances	demand supply feasibility constraints, demand attributes
5	Problem instance	Exact set of demand, supply, constraints and objective	

Table 1: Terminology used in the paper.

strate how they may be reused during evacuation of people and non-living goods (earth). Then we present the RAM semantic model capturing the common information requirements. Next, we use the case-study of work-force matching to show how the ontology may be used. We discuss how the ontology can be consumed and demonstrate its benefits in new situations. We conclude with related work and contributions. To the best of our knowledge, this is the first work on knowledge engineering in resource allocation problems.

Preliminaries

In this section, we clarify the terminology used, illustrate resource allocation problems and motivate the aim of the paper.

Terminology Used

Since resource allocation problems appear in a variety of areas, we clarify the terminology used in the paper in Table 1. Among the terms, *scenario* is peculiar to industry since a collection of problem instances, slightly varying in inputs, are solved together to arrive at a decision for a business situation.

Sample of Allocation Problems

We look at allocation problems in 4 different domains. We use three to build the semantic model and the fourth to test it.

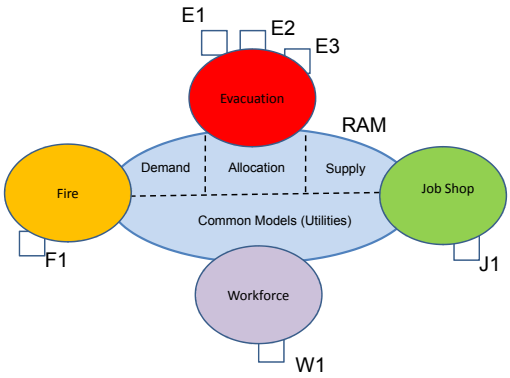


Figure 1: Information hierarchy in allocation problems.

Job Shop: The job shop (scheduling) problem(Graham 1966) is that of efficiently allocating resources to a demand of a set of N jobs ($J = \{j_1, j_2, ..j_N\}$) to the available supply of M machines ($M = \{m_1, m_2, ..m_N\}$). An allocation is an assignment x of J to M. If $C(x)$ represents the cost of an assignment, the allocation problem is to find x with minimum $C(x)$. We will refer to the problem as *Job-Shop*. Variants of the problem differ in the requirements of the job for processing, the capability of the machines and how the cost of assignment is calculated. Although the simplest of the allocation problems considered, it is NP-Hard to solve by directly mapping the known NP-Hard Traveling Salesperson Problem(Applegate *et al.* 2006) to it (let m=1; salesman is the machine and the cities are the jobs).

Workforce Matching: An IT organization is responsible for providing delivery services to its customers across the globe. Here demand is a steady stream of opportunities/projects which can be described with some structured fields such as skills required, duration of the project, cost description, geo-spatial attributes and business restrictions and some unstructured attributes such as good communication skills, development experience should be more than 5 years, etc.(Dixit *et al.* 2009). To meet this demand, they carry workforce varying in their skills, aspirations, preferences, cost and experience levels which evolves continuously as resources move in and out, acquire new skills and experience over time. By allocation, we refer to the matching of certain skilled resources meeting all or some requirements specified in a demand description for a specified duration.

Forest Firefighting: On average, more than 100,000 wildfires, also called wild land fires or forest fires, clear 4 million to 5 million acres (1.6 million to 2 million hectares) of land in the U.S. every year. Uncontrolled blazes fueled by weather, wind, and dry underbrush, wildfires can burn acres of land-and consume everything in their paths-in mere minutes. These fires need to be contained and a tactical plan is created in wake of such a demand. Demand for firefighting resources and equipments can be assessed by forest agencies depending of the various field factors and sent to dispatch locations(Bratten 1970). These dispatch locations are spread across the state and some actor is responsible for allocating type of resources, how many and from where they should be dispatched. The resources can be human fighters as well as non-human resources such as helicopters, dozers, etc. Each resource has a cost, capacity to contain fire, and limitations to work in a demand location.

Evacuation: A common real world problem is evacuation of people and goods in a city. In the former, there are at least two cases discussed in literature - evacuation of people from emergency site in a city(Inampudi and Ganz 2009) and evacuation of a crowd from a building (Wang *et al.* 2008). We will refer to the first problem as *Open-crowd-evacuation* and the second as *Building-crowd-evacuation*, respectively. In the latter, as in construction industry(Ji *et al.* 2010), earth has to be excavated, transported and filled in new locations. One would want to efficiently solve the problems as well

as reuse their common characteristics.

Discussion and Motivation for Modeling

Figure 1 pictorially depicts the information inherent to resource allocation, the relationship among the domain of problems (ovals) and their instances (rectangle). A person solving an allocation problem would have to identify concepts related to demand, supply and allocation. Then, she will have to express constraints relevant to the problem at hand, formulate it in the appropriate language so that their favorite standard or custom tool can solve it.

Our approach is to identify common concepts and relationships, and formally represent them using an *upper-level ontology* called Resource Allocation Model (RAM). RAM is modular, extensible and expressed in the Web Ontology Language (OWL). To solve a specific allocation instance in a domain, a user has to select problem building blocks (e.g., demand), instantiate them with data, review as needed, and then invoke a solver. We have built utilities to streamline the solving process.

RAM helps the user to quickly identify common concepts found useful in other similar problems as well as low-level constraints related to them. Further, the selection tool expresses the selected sub-models in the standard form ready to be consumed. The user still spends time in problem formulation but it is for reviewing the selection and extending the constraints to the new situation. But, time spent now is lower than what she would have had to spend if she had to create the common concepts and constraints from scratch. RAM also reduces hit-and-trial related to naming of concepts (decision variables) because of model reuse.

A note about modeling notation used. Scheduling problems use attribute variables V_i with domain D_i representing the possible values the variable can take. D_i can be discrete or continuous. A constraint is a relation involving k attributes which restricts the domain of values they can collectively take[1]. When representing using an ontology, attribute variables are usually captured as concepts (resources) and their relationships are properties. Constraints are represented as restrictions on relations.

The RAM Semantic Model

In Figure 3, the categories of sub-models inside RAM are shown. Apart from separate categories for allocation, demand and supply, information useful in posing typical problems and common to multiple categories, like skill and geography, are put in a separate utilities category. The left hand side shows the core RAM model while on the right side, extensions needed to solve Workforce domain is highlighted (with arrow).

A closer look at what is modeled and their relationship is shown in Figure 2. *AllocationProblem* and

[1]Special cases are (a) Satisfiability where all variables can take binary values and constraints are clauses, (b) (binary) Constraint Satisfaction Problems where all domains are discrete and all constraints involve two attributes.

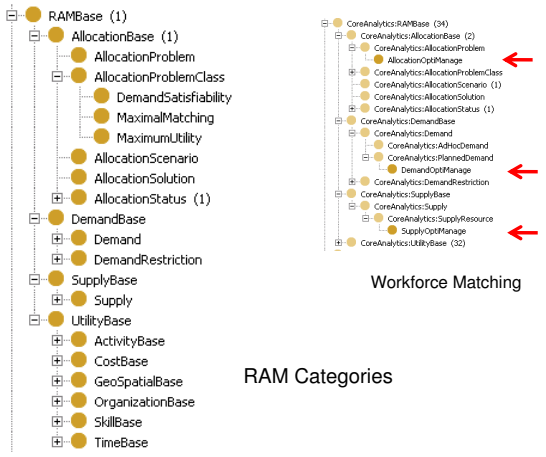


Figure 3: Categories of models in RAM and Workforce.

AllocationSolution are independent concepts, as also are *Demand*, *Supply* and *AllocationScenario*. They are linked by bi-directional relationships (properties). All key concepts are defined in a standardized manner once with rich, necessary, attributes and reused heavily. For example, reuse of *Date* is explicit in the figure while geo-spatial (*City*), skill (*JobRole*) and cost information (*Budget*)are implicit through object properties.

We next discuss the steps involved in identifying common patterns among resource allocation problems and then how they are formally modeled in RAM.

Identifying Common Information Patterns

This section presents the steps involved in identifying the common informational patterns among resource allocation problems identified in various domains. This is important to build a robust semantic model with the aim to reuse it on new domains with little modifications to the core model. The objective is to identify some generic core concepts(attributes) which form the core of the underlying semantic model and some domain specific concepts to complete the domain specific model. The existence of these patterns is not plausible given a well studied common underlying problem,i.e., *resource allocation*. We capture the core concepts for demand, supply and allocation sub-modules which as the literature suggests, are three building blocks for any resource allocation problem.

We follow an authoritative approach to identify the concepts in absence of any standards in this space by referring to the rich existing literature for the past three decades. Modelers have tried to solve resource allocation problem for different domains via multiple algorithmic approaches but the underlying model is approximately invariant. This provides us with enough confidence to extend semantic modeling technology to consume similar analytics in more efficient and less er-

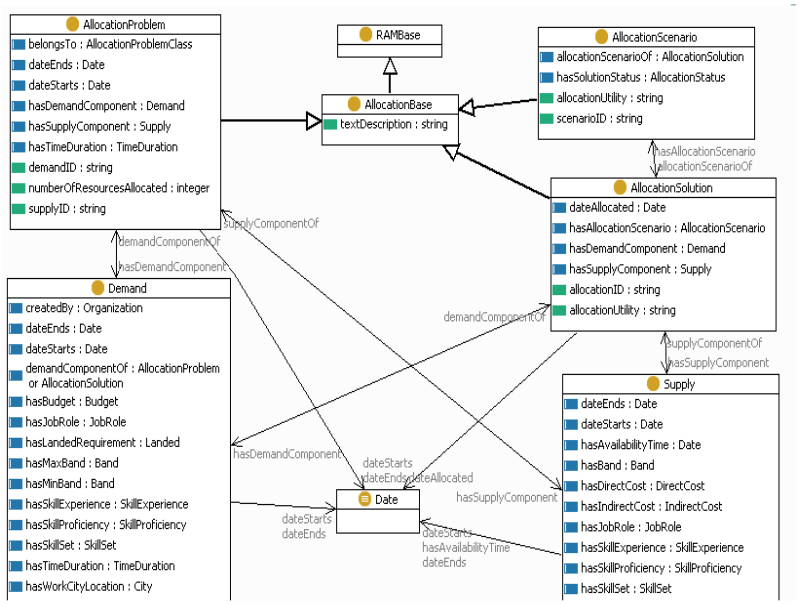


Figure 2: Relationships among key Resource Allocation Model concepts (classes).

ror prone manner. Figures 4, 5, and 6 describe core concepts identified for demand, supply and allocation sub-modules respectively. The tabular format describes the concept identified, the domains in which the concept is realized and in what form, the short abstract description for modelers along with an importance metric. Once we identify the concepts for each sub-module, we test their correctness via *Open-crowd-evacuation* problem assess whether the extraction is generic and robust.

Demand Module

In this subsection we look at the demand sub-module and rationalize the demand concepts identified in detail. For e.g. *Demand.Location* is a core concept. Location in abstract sense refers to any point in space which can be tagged to either an address or landmark or zone or city. There exist detailed geo-spatial ontologies which can be leveraged to get the location description to the desired level for a demand event. This concept is important in mostly all domains as it specifies the work city location for an IT service delivery domain, fire location for a wildland fire management domain and emergency site location for an emergency management domain. Similarly other concepts such as time, duration, criticality, demand for resources etc are of high importance through which one can describe demand to a reasonable extent.

Importance metric has two dimensions; scope and priority. Scope of a concept can be core or additional (domain specific) and priority of a concept given a scope can be high or low. Scope is chosen by the user depending on the coverage of the concept while describing the specific sub-module. For. e.g. location, start time, duration, criticality, resource demand, cost, demand description and ID are selected as core concepts based on their high coverage where as response time and growth rate are domain specific concepts.

We decide the priority based on the potential of an attribute to be part of constraint set for the allocation problem. This assessment requires some domain expertise or a vast review of the kind of resource allocation problems people have been solving in different domains. For e.g. in *Workforce-matching* problem, one objective is to assign resources to demand jobs so that resource utilization is maximized. In this problem, start time becomes a high priority concept. Similarly, for *Fire-fighting* and *Evacuation* problems, the exact location where resource needs to be dispatched is an important attribute. Attributes like response time and travel time derive their values once the demand location and supply location is specified. On the other hand, cost of a demand is a low priority core concept because it does not inhibit the solver to generate a feasible allocation solution. Job-Shop scheduling is one such example where cost assumed is a unit cost as all the jobs are equally important. Another example is *Open-crowd-evacuation* where for each victim there is a unit social cost with the objective to maximize the number of lives saved. Hence demand cost becomes a moot point for the modeler given the allocation choices he can have.

At an aggregate level importance metric depicts our modeling choice on core vs domain specific concepts. Such a categorization is crucial to avoid fragility in the model. However this is just a snapshot of the process involved in identification and may not be complete for modeling purpose. We will show the modeled concept and instances in Section .

Supply Module

Similar to demand, *Supply.Location* becomes part of the core set of concepts. It is important to know where the resource is hosted. For *Workforce-matching* problem, one needs to know if the resource location is compatible with demand location. In *Fire-fighting* problem, it is important to allocate resources from the location where travel time is minimum so that initial attack can be performed as early as possible. For *evacuation* problems, it is desired to call for help from the closest location. Hence resource location becomes a key concept in these set of problems. Similarly, a resource is characterized by when it is available for service hence availability time, what is its capacity and output rate and what kind of skills it possess. For e.g. a helicopter can carry 4 victims and might be available only during the day time with a travel velocity of 150 miles/hr. Along with evacuation skills, it can also be used for creating an aerial fire line during fire fighting process. One can see the high coverage of the above mentioned attributes in nearly all the domains.

On the other hand organization is categorized as domain specific variable. Resources are typically owned by an organization. Although organization is important only when one is leveraging resources from multiple agencies such as in the case of *Fire-fighting* problem. In case where *Workforce-matching* is required for different jobs within an organization, it becomes insignificant. Similarly, in case of emergency management scenario it is not important to know which organization owns the paramedics. Again, such choices are guided by a domain expertise.

Allocation Module

Modeling allocation problems can range from simple models to complex models. This can be characterized via the model output or the number of decision variables in the model. However, looking at some of the domain specific models and within each domain anchoring on a reasonably complex model, we identified some core and additional set of concepts. As emphasised earlier, the set might not be exhaustive to model any domain completely or any specific constraint within a domain; which is not the objective of this exercise. To make sure the core concepts possess a good coverage, we tested some procedural queries asked by modelers quite often as the potential output from a resoure allocation model.

- How many resources are allocated to demand d.
- How many resources are allocated to demand d at location l. (in case of multiple demands competing for resources)
- What type(skill) of resources are allocated to demand d.
- How are allocated resources scheduled to serve demand d.

The set of queries becomes a litmus test for any other allocation model one can think of.

On the other hand, goodness of allocation is categorized as a domain specific concept. Measure of assignment quality is highly domain specific (i.e. cost, quality, jobs matched, etc.) and orthogonal to the existence of an allocation. Optimum allocation or maximal matching is one allocation given the goodness objective hence in absence of such a metric one can still desire a feasible allocation. We also observe that concepts identified are able to capture the output of an *Evacuation* problem fairly well which is later explained in section .

Creating a Reusable Resource Allocation Model

This section presents the mapping of concepts identified in Section to the semantic model built for resource allocation problem. This mapping is not trivial as we aim to build a modular semantic model (Fig: 7) which requires information aggregation in easily extentible and reusable classes. Although, Section identifies demand, supply and allocation as three sub-modules, one can observe that some geo-spatial and temporal concepts are present in all three modules.

Such an observation triggers us to capture these attributes in a separate class called *UtilityBase* along with three other naive classes such as *DemandBase*, *SupplyBase* and *AllocationBase*. *UtilityBase* consists of generic core concepts which are reused by more than one sub-module. For e.g. it consists of *GeospatialBase* which captures information on location. Location can be either an address or a landmark in a region or a zone in a city or a coordinate with latitude and longitude. Existing geospatial ontology(Kim *et al.* 2009) and standards(ISO 2011) are referenced in the model. To use this location description for supply and demand, we define objects like *SupplyGeoDescription* and *DemandGeoDescription* which refer to *GeoSpatialBase* via object properties (Fig 8). Figure 8 depicts the class structure for different objects related to geo-spatial concepts and properties among those objects. For e.g., in this illustration, we restrict the location to a city level. A city *hasCountry* and it can be a *workCityLocationOf* some *Demand* or *Supply*.

Similarly, start-time, duration, and availability time. are temporal concepts which are organized in *TimeBase* under *UilityBase* and corresponding demand, supply and allocation objects are in their respective base as *TimeDescription*. Alogn with that, we have *SkillBase*, *CostBase* and *ActivityBase* as part of the core model. Organization although generic to be part of utility base is an extension to the core model. Similarly, job-role skill-set identified as core-high for all demand, supply and allocation is modeled as core concept in skill base; however it can have some domain specific variations, unlike time and location, Hence, those variations are modeled as extensions in different domain models. For e.g., in IT service domain, a resource is characterized not only by job role and skill type but also with the proficiency and relevant experience in that skill set. We build a core model for resource allocation problem (RAM Base) learning from the core concepts (high and low) identified and show a subsequent extension of this core model when we instantiate the *Workforce-matching* problem.

Figure 9 shows a snapshot of the demand and supply instance graphs (upper part) generated using the TopBraid Composer tool (a licensed tool for semantic modeling) as well as their details in a table format (lower part) for *Workforce-matching* problem. We show that extensions required for the domain can be easily added on the core RAM model. However, it is interesting to note that core model covered 80% of demand attributes and 60% of supply attributes.

Allocating Resources in Workforce Domain

In this section we illustrate the benefits of allocating resources in workforce domain using semantic models by comparing modeling experience of a modeler with and without RAM. Table 2 lists 7 essential steps required for modeling and solving a new resource allocation problem in any domain. It illustrates how using knowledge engineering to create reusable optimization semantic models can reduce error and complexity from modeling steps which are complex and error prone due to manual encoding involved in them.

Allocating With RAM

Table 2 shows that having a reusable model (i.e. RAM) for resource allocation problem is useful in reducing the complexity and error susceptibility. Modeling with RAM, the effort of the modeler will be directed towards reviewing things rather than discovering and encoding them. RAM models core components in resource allocation problem which serves as a platform for users to quickly model, solve and compare resource allocation problems in any domain. Useful metrics such as complexity of the problem, solution time and solution status are modeled in allocation module which allow users to generate deeper insights while comparing different scenarios in RAM. Figure 10 describes one of the potential process flow a modeler goes through while solving allocation problem. Given a problem like *Workforce-matching* or *Open-crowd-evacuation*, the user can browse the RAM model and select sub-models of interest in DemandBase, SupplyBase and AllocationBase. The SelectionTool instantiates the selected models into standard solution formats. One example is MPS (Mathematical Programming System(Wikipedia 2011)) which is a file format for presenting and archiving linear programming (LP) and mixed integer programming problems. It is supported by leading solvers.

The user can review the generated candidate formulation, review it and enhance it. One area which needs review and enhancement depending on the new problem is constraints. Even in a problem category like evacuations, constraints can vary based on whether people (living) are being moved or earth (non-living), the people are being moved in a building or in an open areas, and so on. However, the concepts between which the constraints are defined remains stable. The finalized models can be run by the standard solver. In the Figure, we show solving of ILP/LP formulations, but can be easily extended to other approaches (e.g., SAT).

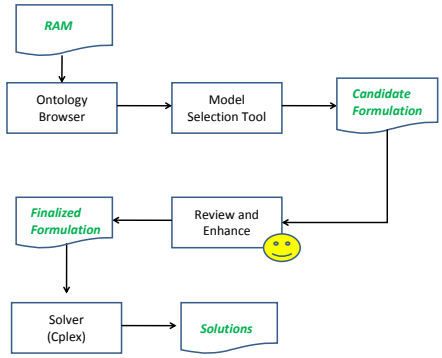


Figure 10: Process-flow to explore *Resource Allocation Model* for a problem, select sub-models of interest, review and solve them using a standard solver.

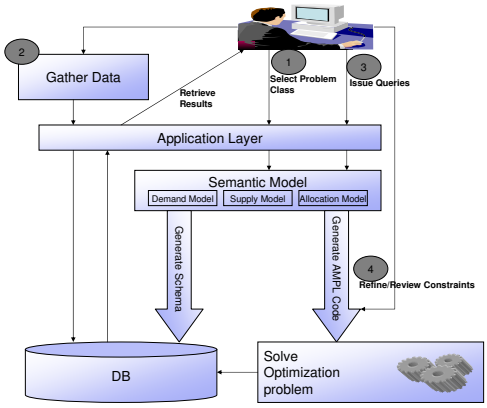


Figure 11: Architecture of a system to advise on allocation problems using RAM.

Steps	Modeling w/o RAM	Modeling w/ RAM	Complexity Changed	Error Suscepti--bility changed
Gather Business Requirements	Required: Needs to be repeated for every domain	Required: Needs to be repeated for every domain	Unchanged	Unchanged
Gather data for problem Instance	Required: Needs to be repeated for every Domain/Instance	Required: Needs to be repeated for every Instance	Unchanged	Unchanged
Generate Date Schema	Required: Needs to be repeated for every Domain	Not Required: Model generates schema automatically which is reused for every domain	High ⇒ Low	High ⇒ Low
Define Model Parameters,decision var., constraints & objective	Required: Modelers build models from scratch for every new domain	Not Required: Core parameters, objective and some constraints are encoded	High ⇒ Low	High ⇒ Low
Write Optimization encoding (e.g AMPL Code)	Required: New Optimization encoding is done for every domain	Not required: Generated automatically from the semantic model at run time based on the parameters selection. Modeler has to review some constraints	High ⇒ Low	High ⇒ Low
Call Solver & generate results	Required: Problem instance is solved every time and results	Required: Automatically Solved. Solver options can be configured.	High ⇒ Low	High ⇒ Low
Write Queries to view results	Required: Write queries every time for every domain	Not Required: Pre-canned queries can be defined and used	High ⇒ Low	Unchanged

Table 2: Modeling Experience using Semantic Models.

Figure 11 describes our proposed architecture and methodology for solving resource allocation problem in a new domain. Below are steps which a modeler has to perform manually to solve the problem.

1. Select the problem class one wants to solve based on the business problem.e.g. resource allocation problem.(Data schema is generated automatically in the back ground for model components such as demand, supply and allocation).
2. Gather data for instance to solve e.g. demand and supply
3. Issue queries (some generic queries can be pre-canned). AMPL code is generated from the semantic model (one time) in the background.
4. Review/refine the set of constraints and continue with solve. Results are displayed back.

We discuss the productivity gains and our experimental results in next section.

Discussion

This section discusses experimental set up, results of workforce matching problem via semantic queries, and productivity gains when a modeler uses RAM for workforce matching problem. We assume productivity gains will increase linearly as we reuse this model for other domains or increase the instance size in terms of number of variables and constraints.

Experiment We formulate a sample workforce matching problem with 16 variables (15 binary and 1 linear), 8 linear constraints and 1 linear objective using the methodology proposed earlier. The sample problem consists of 4 demand requests and 4 supply resources described in Table 3 & 4 respectively. Demand request has 10 demand attributes and supply resource has 7 supply attributes out of which job-role, skill-set, city location and band constitute the matching attributes set. The problem is solved for maximum allocation utility. This sample instance is extracted from (Dixit *et al.* 2009).

Model Components	LOC to write		LOC to review	
	w/o RAM	w/ RAM	w/o RAM	w/ RAM
Parameters	120	5	120	115
Decision var.	3	0	3	3
Constraints	5	5	5	5
Objective	8	0	8	8
Configuration	65	15	65	50
Total	201	25	201	181

Table 5: Statistics about AMPL code (without and with RAM).

We load demand and supply data in RAM and generate optimization code (i.e. AMPL formulation). To generate model formulation, we need some preprocessing to compute feasible supply for each demand request and vice-versa based on a predefined matching criteria (hard constraint), and utility for each possible allocation based on soft constraints. This is done at run time using semantic queries. Modeler only needs to review the formulation, enhance the constraint set, and solve the problem. The allocation results are loaded in RAM from where it is retrieved using generic predefined semantic queries for resource allocation problems. First we illustrate the productivity gains in terms of lines of code one has to write v/s. review with and without RAM. We use the code generated via RAM as our base for comparison. Table 5 captures this comparison for the sample problem. Notice that the lines of code to write reduces by nearly 90% while that to review reduces by 10%.

One can extrapolate the benefits when we reuse the model for solving multiple resource allocation problems in the same or other domains. RAM can thus hot start the modeling process and reduce complexity and error susceptibility.

Next we illustrate some important queries and their corresponding results set from the problem instance at hand. We use TopBraid (a licensed software for modeling ontologies) to model RAM and query ontologies. Figure 12 captures the SPARQL query and the corresponding result set from the implementation. We discuss the result set below to ensure that the problem was

Demand--ID	Start Date	End Date	Job Role	Skill Set	Min Band	Max Band	City	Country	# open positions
10605	10/24/2011	12/31/2013	Project Manager	Project Office Management	6B	7B	Bangalore	India	1
32162	11/23/2011	11/23/2015	Application Developer	COBOL	6B	7B	Bangalore	India	3
32158	11/23/2011	11/23/2015	Project Manager	Project Office Management	6B	7B	Bangalore	India	1
5771	10/24/2011	3/30/2012	Project Manager	Project Office Management	6B	7B	Bangalore	India	1

Table 3: Demand data for sample workforce matching problem.

Demand--ID	Availability Date	Job Role	Skill Set	Band	City	Country
27912	11/16/2011	Application Developer	COBOL	6B	Bangalore	India
23763	1/2/2012	Project Manager	Project Office Management	7B	Phoenix	US
31593	1/2/2012	Project Manager	Project Office Management	6B	Bangalore	India
29380	1/2/2012	Project Manager	Project Office Management	6B	Bangalore	India

Table 4: Supply data for sample workforce matching problem.

solved as desired.

- ***Query1:*** Find Demand Gap
  Comments: Out of 4 demand requests, 3 were allocated resources. However demand '32162' required 3 resources but only 1 was allocated, hence causing a gap of 2.
- ***Query2:*** List all Unallocated Supply
  Comments: Out of 4 supply resources, 1 resource '23763' was not feasible for any demand due to location constraint. Rest all were allocated.
- ***Query3:*** List all allocations and their utilities
  Comments: 3 assignments were made for the problem under the defined matching logic. Allocation Utility is computed based on the difference in the availability date of a resource and start date of the demand using a function pre-defined by the modeler.

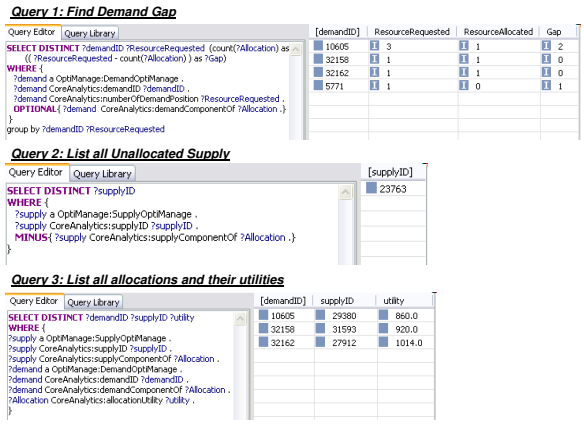


Figure 12: Pre-canned Semantic Queries in RAM

We experiment with scenarios where different matching attributes are used and compare metrics such as number of assignments, total utility, etc. Such comparisons are useful for modeler to understand tradeoffs between quality of allocation and quantity of allocation. Using RAM, modeler can run many *'What-if'* scenarios and not only compare results but also their complexities and solution time. One can use two well studied techniques to reason with the model - semantic search(Guha *et al.* 2003; Hildebrand 1998) and model comparison(Mandelin *et al.* 2006). In the former, semantic annotations from the model are used in various stages of a search algorithm to return highly accurate results. In the latter, two models can be compared based on syntactical as well as semantic attributes in the models. State-of-the art tools like ILOG ODM provides the interface for similar *'What-if'* analysis however it is limited to problem instance comparison in one domain only and does not let user gain insights about problem complexity.

Discussion and Related Work

Until now, we analyzed common patterns from select allocation problems, represented them in an ontology and showed how they could be consumed in *Workforce*. Now, we show reuse of models in our test domain of evacuation and put this work in context of related work.

Reusing Models in a New Domain

Now consider the problem of evacuating people from emergency scenes in a city (*Open-crowd-evacuation*(Inampudi and Ganz 2009)). One could use the ontology browser and look for models in RAM. Table 6 gives a high-level mapping of the allocation, demand and supply aspects of the problem. It is significant to note that the RAM model was not created with this problem. However, we could cover 6 of the 7 problem characteristics using RAM, this demonstrating that there is major scope for reuse – hiter-to not done in state-of-the-art. Besides, RAM model can evolve with new

S.No.	In(Inampudi and Ganz 2009)	In RAM	Comments
1	Quantity of resource to be allocated from Depot i to Emergency Site j	*Allocation.ResourceAllocated*	In RAM with *Allocation.Location*
2	Number of resources requested from Emergency Site j	*Demand.ResourceDemand*	In RAM
3	Priority for resource at Emergency Site j	*Demand.Criticality*	
4	Total number of resources available at depot i	*Supply.ResourceSupply*	RAM allows type of resource also
5	Time taken to transfer resource from depot i to Emergency Site j per transportation vehicle	*Supply.OutputRate*	Real time or estimated
6	No of transportation vehicles at depot i	*Supply.ResourceSupply*	RAM allows type of vehicle also
7	No of resources per transportation vehicle	*Supply.Capacity*	In RAM

Table 6: Reusing RAM sub-models in a new setting - crowd evacuation in open.

concepts and reuse can further increase with newer domains. Highlighted column in Figure 4 reinforces the selection process based on the other three allocation problems. However it is possible to promote or demote a concept from core to additional and vice-versa as more problem descriptions are added to the repository.

We also did mappings for creating instances in related domains of *Building-crowd-evacuation* and earth moving. We found reuse of concepts ranging from 30% to 100%.

Related Work

The allocation of resources to demand over time is a fundamental problem that is central to Management Science, Computer Science and Economics. The questions investigated by computer scientists are often of a procedural nature (how do we find an allocation?), while economists are more likely to concentrate on qualitative issues (what makes a good allocation?). Both the queries are applicable to wide range of application domains such as manufacturing and scheduling(Carlsson *et al.* 1999), airport traffic management(Jonker *et al.* 2005), crisis management(Inampudi and Ganz 2009), transportation and logistics(Sandholm 1993) and workforce management (Dixit *et al.* 2009).

Allocation problems are related to AI Planning where knowledge engineering has been of immense interest in recent years. In fact, there is a competition track at ICAPS conferences on it (ICAPS-KE 2011). However, the modeling is planning centric and does not cover allocation problems in significant detail. Operations Research (OR) community, on the other hand, has not realized the importance and benefits of knowledge engineering, which this paper demonstrates.

Conclusion

In this paper, we considered the problem of how to efficiently create allocation problems by reusing sub-models from similar problems. This helps us to reduce the time needed to formulate new allocation problems, reduces hit-and-trial and focuses the user to the most important issue in problem setup – review and provide constraint level new information. We analyzed patterns of demand, supply and allocation in three diverse areas of job-shop, workforce-matching and fire-fighting. We created a semantic model called RAM, and showed that it can be reused for considering the new domain of evacuation problems. Then using code-generation utilities for standard solvers and semantic queries, we show that RAM can expedite problem solving for resource allocation problems. To the best of our knowledge, this is a first effort to organize and reuse models for allocation problems.

References

AMPL. A modeling language for mathematical programming. In *http://www.ampl.com/, last accessed Dec 16, 2011.*, 2011.

D. L. Applegate, R. M. Bixby, V. Chvtal, and W. J. Cook. The traveling salesman problem. In *ISBN 0691129932*, 2006.

J. Christopher Beck and Mark S. Fox. A generic framework for constraint-directed search and scheduling. In *AI Magazine, Winter*, 1998.

T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities. In *Scientific American, May, pp. 28-37*, 2001.

F.W. Bratten. *Allocation model for firefighting resources– a progress report.* Research note PSW. Forest Service, U.S. Dept. of Agriculture, Pacific Southwest Forest and Range Experiment Station, 1970.

Mats Carlsson, Per Kreuger, and Emil Astrm. Constraint-based resource allocation and scheduling in steel manufacturing. In *PADL'99: First International Workshop on Practical Aspects of Declarative Languages, San Antonio, Texas, USA.*, 1999.

Kashyap Dixit, Munish Goyal, Pranav Gupta, Nanda Kambhatla, Rohit M. Lotlikar, Debapriyo Majumdar, Gyana R. Parija, Sambuddha Roy, and Soujanya Soni. Effective decision support for workforce deployment service systems. *IEEE Intl Conf on Services Computing (SCC)*, 0:104–111, 2009.

R. Graham. Bounds for certain multiprocessing anomalies. In *Bell System Technical Journal 45: 15631581*, 1966.

R. Guha, R. McCool, and E. Miller. Semantic search. In *Proc. WWW. Available at: http://www2003.org/cdrom/papers/refereed/p779/ess.html*, 2003.

Hildebrand. Semantic search survey. In *Wikipedia entry - http://swuiwiki.webscience.org/index.php/ Semantic_Search_Survey, last accessed June 23, 2011.*, 1998.

T. Ibaraki and N. Katoh. *Resource allocation problems: Algorithmic approaches.* MIT Press series in the

foundations of computing. Cambridge, Mass. : MIT Press, 1998.

ICAPS-KE. Knowledge engineering for planning and scheduling. In *http://icaps11.icaps-conference.org/workshops/keps.html, last accessed June 23, 2011.*, 2011.

ILOG. Ibm ilog cplex. In *Wikipedia entry - http://en.wikipedia.org/wiki/CPLEX, last accessed Dec 16, 2011.*, 2011.

Venkata S. Inampudi and Aura Ganz. Web based tool for resource allocation in multiple mass casualty incidents. In *31st Annual International Conference of the IEEE EMBS, Minneapolis, USA, September 2-6*, 2009.

ISO. 3166-1: English country names and code elements. In *http://www.iso.org/iso/english_country_names_and_code_elements, last accessed June 23, 2011.*, 2011.

Y. Ji, A. Borrmann, E. Rank, F. Seipp, and S. Ruzika. Mathematical modeling of earthwork optimization problems. In *Proc. Internation Conference on AComputing in Civil and Building Engineering, Nottingham*, 2010.

Geert Jonker, John jules Ch. Meyer, and Frank Dignum. Efficiency and fairness in air traffic control. In *Belgium-Netherlands Conf. on AI*, pages 151–157, 2005.

S. Kim, M. I. Sucasas, c. Caracciloi, and J. Keizer. Integrating country-based heterogeneous data at the united nations: Fao's geopolitical ontology and services. In *http://semanticweb.com/integrating-country-based-heterogeneous-data-at-the-united-nations-fao-s-geopolitical-ontology-and-services_b10681, last accessed June 23, 2011.*, 2009.

David Mandelin, Doug Kimelman, and Daniel Yellin. A bayesian approach to diagram matching with application to architectural models. In *Proc. 28th Intl Conf. on Software engineering*, ICSE '06, pages 222–231, New York, NY, USA, 2006. ACM.

M. Moskewicz, C. Madigan, Y. Zhao, L. Zhang, and S. Malik. Chaff: Engineering an efficient sat solver. In *39th Design Automation Conference (DAC 2001), Las Vegas.*, 2001.

Jeff Riley. Conceptual graph representation of a work order allocation system. In *CS Department, Royal Melbourne Institute of Technology, Web References http://environment.nationalgeographic.com/environment/natural-disasters/wildfires/*, 1996.

T. W. Sandholm. An implementation of the contract net protocol based on marginal cost calculations. In *Proc. 11th National Conference on Artificial Intelligence (AAAI), pages 256-262. AAAI Press*, 1993.

Peng Wang, Peter B. Luh, Shi-Chung Chang, and Jin Sun. Modeling and optimization of crowd guidance for building emergency evacuation. In *4th IEEE Conference on Automation Science and Engineering Key Bridge Marriott, Washington DC, USA*, 2008.

Wikipedia. Mathematical programming system. In *Wikipedia entry - http://en.wikipedia.org/wiki/MPS_(format), last accessed June 23, 2011.*, 2011.

Attributes/ Concepts	Importance	Job Shop Scheduling	Workforce - matching	Fire-Fighting	Open Crowd Evacuation	Description
Location	Core-High	X	√ (Work City Location)	√ (Fire Location)	√ (Disaster Location)	Location at which any event triggers the demand of resources. It could be a city, a point, a country, an address etc
Start Time	Core-High	X (output metric)	√ (Start Date)	√ (Fire Start Time)	√ (Incident Time)	Time (Date) at which the event started or going to start
Duration (End Time)	Core-High	√ (Processing Time)	√ (Project Duration)	X (metric to be minimized)	X (metric to be minimized)	Duration by when the demand needs to be completed
Criticality	Core-High	X	√ (Time to Start)	√ (Dispatch Level)	√ (Resource Priority)	Urgency to respond to a demand
Resource Demand	Core-High	√ (# machines)	√ (Number Positions)	√ (# suppression resources)	√ (Resource Demand)	Number of "unit" resources required.
Demand ID	Core-Low	√ (Job ID)	√ (Seat ID)	√ (Fire ID)	√ (EMS ID)	Unique Identifier for demand
Demand Description	Core-Low	√ (Job Description)	√ (Job Description)	√ (Event description)	√ (Event description)	Text description for an event triggering a demand
Cost	Core – Low	√ (Unit Cost)	√ (Band Requirement)	√ (acreage burnt)	X	Abstract concept to measure the effectiveness of output
Response Time	Additional - High	X	X	√ (response time)	√ (response time)	Time required to respond i.e. time by when the demand will be lost
Growth Rate	Additional - Low	X	X	√ (rate of spread)	X	Rate at which incident demand grows if demand is not static

Figure 4: Core concepts for demand module. √ denotes that the concept is relevant to the problem while X denotes that it is irrelevant. Highlighted scenario is explained in Section .

Attributes/ Concept	Importance	Job Shop Scheduling	Workforce - matching	Fire-fighting	Open Crowd Evacuation	Description
Location	Core - High	X	√ (Work City Location)	√ (Fire Planning Location)	√ (Resource Location)	Location at which resources reside for demand assignment/dispatch.
Availability Time	Core - High	√ (Availability Time)	√ (Availability Date)	√ (Availability Time)	√ (Availability Time)	Time (Date) at which the resource is available for dispatch
Capacity	Core - High	x	√ (Shift Length)	√ (Fire Fighting Capacity)	√ (Vehicle Capacity)	Capacity of a resource to perform in units of time, or carrying capacity
Output Rate	Core - High	√ (processing time)	X	√ (Fire Production rate)	√ (Resource Velocity)	Rate to perform a unit job.
Resource Supply	Core-High	√ (# machine)	√ (# professionals)	√ (# resources)	√ (# ambulances)	Number of "unit" resources of type
Job role -Skill Set	Core - High	X	√ (Job Role Skill Set)	√ (Resource Type)	√ (Resource Type)	Type of Resource
Supply ID	Core - Low	√ (machine ID)	√ (resource ID)	√ (Resource ID)	√ (Resource ID)	Unique Identifier for supply
Cost	Core - Low	x (Machine Cost)	√ (Band)	√ (Resource Cost)	X	Abstract concept to measure the cost of resource
Resource Description	Core -Low	√	√	√	√	Text description for a resource
Organization	Additional – Low	X	√ (Business Unit)	√ (Fire Planning Unit)	√ (Hospital)	Organization associated or owning the resource

Figure 5: Core concepts for supply module. √ denotes that the concept is relevant to the problem while X denotes that it is irrelevant. Highlighted scenario is explained in Section .

Attributes/ Concepts	Importance	Job Shop Scheduling	Workforce Matching	Fire-fighting	Open Crowd Evacuation	Description
Location	Core - High	X	√	√	√	Location at which resource will be positioned to serve the demand, it can be demand or supply or some remote location
Start Time	Core - High	√	√	√	√	Time (Date) at which the resource is scheduled to serve the demand.
Duration (End Time)	Core - High	√	√	X	X	Duration for which the resource will be assigned and cannot be used by anything else.
Job role -Skill Set	Core- High	X	√	√	√	Type of resource allocated
Resource Allocated	Core - High	X	√	√	√	Number of Resource Allocated
Assignment Time	Core - Low	X	√	√	X	Time at which the dispatch decision is made
Supply ID	Core - Low	√	√	√	√	Identifier for supply allocated
Demand ID	Core - Low	√	√	√	√	Unique Identifier for demand
Goodness of allocation	Additional-Low	√	√	√	√	Assignment cost or quality

Figure 6: Core concepts for allocation module. √ denotes that the concept is relevant to the problem while X denotes that it is irrelevant. Highlighted scenario is explained in Section .

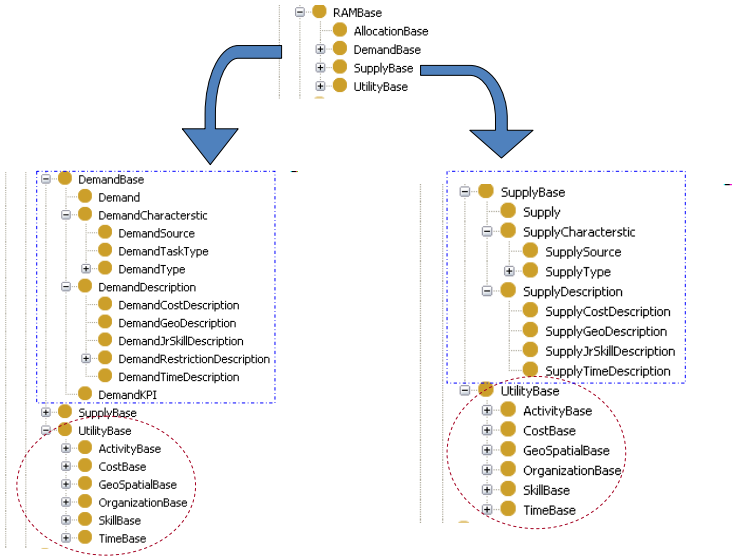


Figure 7: Resource Allocation Model

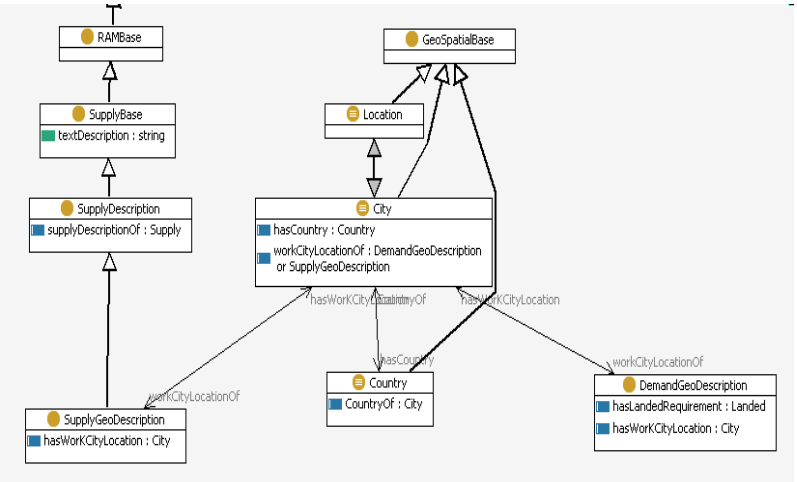
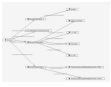


Figure 8: Semantic Model Representation for Geo-Spatial Base



Workforce-matching Demand Instance

Concepts in RAM	Demand Attributes	Sample Demand Instance
Location (Core)	Location country	India
Location (Core)	Location City	Bangalore
Resource Demand (Core)	Number needed	1
Demand ID (Core)	Demand ID	390544
Job Role (Core)	Job Role	application developer
Skill Set (Core)	skill set	TSO_ISPF
Start Time (Core)	Start Date	10/3/2008
End Time (Core)	End Date	12/31/2009
Cost (Core)	Min Band	7
Cost (Core)	Max Band	8
Demand Description (Core)	Job Description	Text
Proficiency (Extension)	Skill Proficiency	Intermediate
Experience (Extension)	Skill Experience	2 years

Workforce-matching Supply Instance

Concepts in RAM	Supply Attributes	Supply Instance 1
Location (Core)	Location Country	US
Location (Core)	Location City	Phoenix([Lat, Long] – [33.48-112.07])
Supply ID (Core)	Professional ID	4D4416897
Job Role (Core)	Job Role	Strategy Consultant
Skill Set (Core)	Primary Skill set	Operations Strategy
Availability Time (Core)	Availability Date	6/30/2010
Cost (Core)	Band	8
Proficiency (Extension)	Skill Proficiency	Intermediate
Proficiency (Extension)	Skill Experience	2 years
Organization (Extension)	Organization	IBM
Name (Extension)	Name	BERKSHIRE; EL

Figure 9: *WorkForce-matching* sample problem instance extracted from (Dixit *et al.* 2009). The green rows represent core while the red rows represent *Workforce-matching* specific attributes.