

# Modeling Hierarchical Resources Within a Unified Ontology

## A Position Paper

Alexander Schiendorfer<sup>1</sup>, Yves Wautelet<sup>2</sup> and Wolfgang Reif<sup>1</sup>

<sup>1</sup>*Institut für Software & Systems Engineering, Universität Augsburg, Augsburg, Germany*

<sup>2</sup>*Faculteit Economie & Management, KU Leuven, Leuven, Belgium*

**Keywords:** Resource Management, Resource Allocation, Unified Ontology, Smart Grids, Hierarchical Modeling.

**Abstract:** Resource-Intensive Software Ecosystems (RISE) can mainly be found in production management but also in virtually any socio-technical environment. RISE appear prominently in the form of *smart grids* or *cloud environments* where optimizing resource utilization and allocation becomes the most important aspect for competitive service provision. In such a context, the need for unified ontologies supported by adaptive software (i.e., software able to learn from and act on its environment) is highly attractive. Indeed, resources are mostly not monolithic entities but active and collaborative agents often organized in a hierarchical manner. A hierarchy implies multiple levels of abstraction leading to resource allocation on different levels of organization – with abstractions being relevant for both inter- and intra-organization resource management. Once adequately defined, the use of constraint-based optimization algorithms on those multiple levels can provide efficient resource allocation. We apply, in this paper, ontological elements to model resources in a unified manner on multiple levels onto an example taken from distributed energy management. Then we present algorithmic ideas to organize the hierarchy of these resources.

## 1 INTRODUCTION

We define Resource-Intensive Software Ecosystems (RISE) as environments where resources take a prominent role in the realization of the services provided to the end-user. In other words, they are crucial for the proper execution of business processes. (Wautelet et al., 2012) presents a unified ontology for the modeling of resources into RISE; within the ontology, resources are modeled in function of their usage (see Section 2). Explicitly modeling resources in a unified and standardized manner (i.e. independent of their *type*, *domain* and *interface*) can provide us with flexible and reusable components.

(Cabanillas et al., 2013) identifies a gap between resource assignment at design time and resource allocation at run time; this paper is part of the willingness to dispose of a tool to make resource allocation at run time only on the basis of resources designed uniformly on the basis of a same ontology. This thus helps filling this gap.

Resource allocation is indeed a task frequently encountered in industrial environments and often resolved through agent-based software solutions implemented within a distributed context (Chevaleyre et al.,

2006; Seebach et al., 2010). It can be achieved by the use of different levels of organization leading to a synthesis and abstraction problem in order to deal with each level and impact on the other levels accordingly (see (Schiendorfer et al., 2014)).

This paper illustrates the use of hierarchical resource composition as depicted in (Schiendorfer et al., 2014) with the ontology presented in (Wautelet et al., 2012). The main contribution is to show the representation is feasible and to explore the benefits of a disciplined resource modeling process that leads to algorithmic advantages in terms of hierarchies. Also by resource composition, we address inter-organizational problems in an efficient manner. For instance, a virtual power plant is a power plant acting on behalf of real power plants. For an energy provider it is thus much more convenient to sell a single value at a certain time to their contractors (managing yet another set of physical plants) which they in turn redistribute. Useful abstract models leaving aside organizations' internal details have to be developed for this purpose.

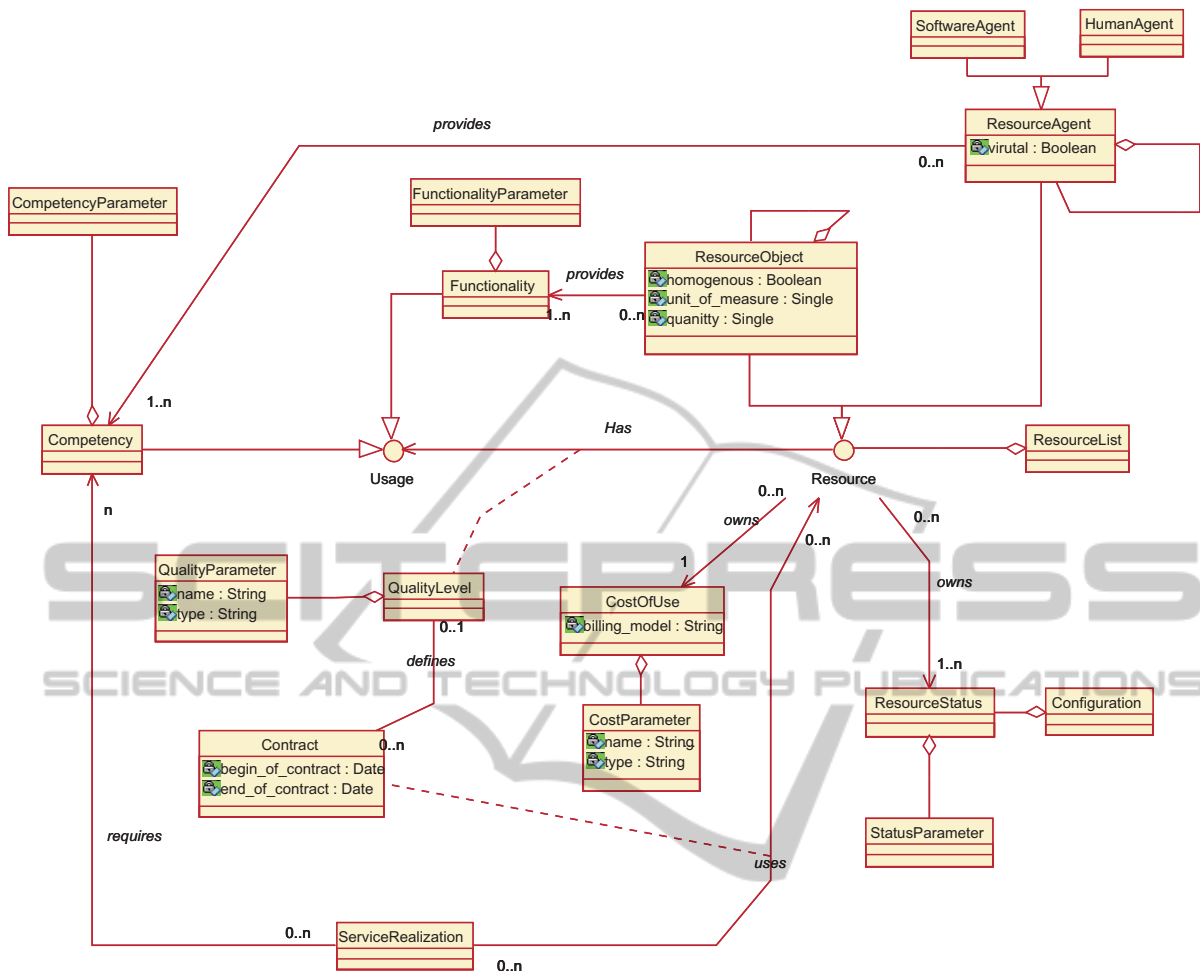


Figure 1: A meta-model representing an ontology for resource representation.

## 2 ABOUT THE UNIFIED-RESOURCE ONTOLOGY

There are numerous definitions of the concept of *resource*. The Merriam-Webster dictionary refers to it as *a person, asset, material, or capital which can be used to accomplish a goal*. While remaining basic, this definition has the interesting aspect of identifying the fact that a resource is not a process (something functional/operational) but should be used for the proper achievement (the goal) of such a process through action(s). This consequently leads to the intuition that resources do own particular skills that are *required* in a functional context (for the realization of processes or services). Within a pool of resources, the requesting service is thus only interested in resources helping it achieving its functional requirements. Nevertheless, depending on their type, resources are *inan-*

*imate* and do have a functional utility (this is the case of *Resource Objects*) or are able to *behave* (this is the case for *Resource Agents*). Resources' offer and demand can consequently not be centralized into a single concept so that *Resource Objects* are *functionality* providers while *Resource Agents* are *competency* providers. Resources *offer* functionalities and competencies while services *demand* them.

In (Wautelet et al., 2012), we model resources as components encapsulating functionalities and competencies and illustrate it on a case study in the steel industry. Figure 1 depicts the unified meta-model using a UML class diagram (OMG, 2011). Details about the concepts and their interdependencies can be found in (Wautelet et al., 2012). The present paper focuses on the hierarchical resource allocation problem within this ontology rather than on functionality and competency modeling.

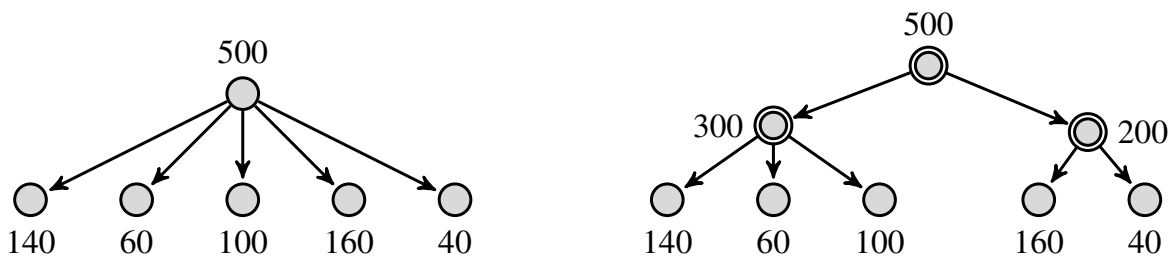


Figure 2: Resource allocation problems can be solved using a hierarchical decomposition structure. Inner nodes representing virtual agents are marked by double circles.

### 3 DEFINING THE HIERARCHICAL RESOURCE ALLOCATION PROBLEM

The ontology presented in (Wautelet et al., 2012) induces a possible hierarchy among resources in the sense that a *ResourceObject* or a *ResourceAgent* can either be an *assembly* – i.e., a resource made of other resources advertising functionalities/competencies – or be *atomic* – i.e., not made of other resources advertising functionalities/competencies. This is represented into the meta-model of Figure 1 by the composition association on the *ResourceObject* and *ResourceAgent* concepts themselves.

A *Resource Agent* taking the form of an assembly that is issued of the combination of other resource agents is called a *Virtual Agent (VA)*. The VA then manages the tasks of its subordinates representing an organization. This is shown in the ontology by the boolean field *virtual* in the *ResourceAgent* class.

As it is important for the case study we also highlight the case of an *homogeneous ResourceObject*. Indeed, a resource like electrical energy is said to be homogeneous because it cannot be counted but is the available in a defined *quantity*. Within the case study of Section 4, a quantity  $x_R$  of an homogeneous resource is to be provided by  $n$  providers  $x_i$  such that  $\sum_{i=1}^n x_i = x_R$ . Figure 2 shows this decomposition and how a certain quantity of such a resource can be hierarchically provided.

### 4 CASE STUDY: ENERGY DISTRIBUTION IN A VIRTUAL POWER PLANT

In conventional energy systems, a small number of rather large and inflexible power plants – such as coal or nuclear power plants – come up for the majority

of the power production. Environment considerations as well as the limited availability of fossil fuels in the future motivate to use a larger number of smaller but renewable energy providers. Examples of the latter providers are wind, hydro, photovoltaic or biogas power plants. The resulting complexity of the system puts stress on the control of the production. Indeed, the deviation between demand and production must be coordinated to leave the utility frequency in a well-defined band (e.g., [49.8, 50.2] Hz in Europe).

Clearly, the higher coordination demand between production and demand is a resource-intensive process and calls for IT-based solutions: the so-called “Smart Grid” systems (Ramchurn et al., 2012). A first step in this direction is the introduction of “Virtual Power Plants”, i.e., a single energy provider takes the responsibility of providing a certain amount of energy at specific time steps by issuing a contract at an energy market<sup>1</sup>. This top-level service, *provide (customers with) energy at defined cost and time slots*, is divided into sub-services that the contractors (which can be organizations or plant providers such as farmers) offer. The contracts are typically traded one day in advance and are binding for all involved parties.

Implementing this situation results in solving an optimization problem, the objective being to allocate energy to the subordinates as cost-effectively as possible such that the amount to be traded at the energy market is met. This corresponds to the general one-good resource allocation problem without externalities (Van Zandt, 1995), i.e. given a total quantity  $x_R$  of a resource, find an allocation  $\langle x_1, \dots, x_n \rangle$  of the resource to  $n$  agents:

$$\text{minimize}_{\langle x_1, \dots, x_n \rangle} \sum_{i=1}^n c_i(x_i) \quad \text{subject to} \quad \sum_{i=1}^n x_i = x_R$$

We illustrate how the unified-resource ontology can be mapped to this scenario. Concretely, we pick the service that a virtual power plant vp has to offer

<sup>1</sup>see, e.g., <https://www.regelleistung.net> for the German energy market

when committed to a certain set of operating reserve for the following day. Assume that we bid to sell 500 KW at 14:00, 600 KW at 14:15, 700 KW at 14:30, and 400 KW at 14:45. For sake of simplicity, these quantities representing the bid for one hour of production are represented as vectors. In total, this amounts to a provided power of 2200 KWh resulting in costs of 3300 € at 1.5 € per KWh. Assume that *vp* has two plants available: a biomass power plant *bm*, and a hydro plant *hy*. All power plants, virtual or not, are represented by the concept *ResourceProvider* which is a specialization of a resource agent that contains a set of intervals *feasReg* for the feasible regions that a resource provider can contribute in.

As discussed, the aim of the paper is to highlight the representation of resources on multiple levels of abstraction. Figure 3 shows two levels of granularity within the unified ontology, of course there could virtually be an infinity of levels. The service to be realized is the service *Provide energy at defined cost and time slots*. The contract

**Service Realization** Provide energy at defined cost and time slots.

**Input** contract:  $\langle 500, 600, 700, 400 \rangle$  at 14:00 at costs  $\leq 3300$  €.

*bm* : *ResourceProvider*    *hy* : *ResourceProvider*

**Requires** Resource  $e_{vp}$  500 at 14:00,  $e_{vp}$  600 at 14:15, etc.

The *bm* and *hy* power plants can firstly be defined as *ResourceAgent*. We define:

**Resource** *ResourceProvider* *bm*

**Type** Resource Agent

**Attribute** feasible Regions :  $[0, 0], [100, 400]$   
costs : 1.7 € per KWh

Having the  $[0, 0]$  interval present indicates that the agent can be disconnected from contributions and if it does contribute, it has to do so above a minimal level. The same description can be made for the *ResourceProvider* *hy*. The feasible regions and costs for that other power plant (*hy*) is given by  $\{[0, 0][50, 300]\}$  at 1.2 €.

Next, we define the energy resource provided by a plant as *ResourceObject*, more specifically:

**Resource** Energy  $e_{bm}$

**Type** Resource Object

**Attribute** quantity and unit

**Provides** Electrical Energy

The same description can be made for Energy  $e_{hy}$ . Finally, we illustrate the aggregation of the resources like:

**Resource** *AggregateEnergy*  $e_{vp}$

**Type** Resource Object

**Provides** Energy equal to  $\sum_{a \in \{bm, hy\}} e_a$

**Resource** *VirtualPlant* *vp*

**Type** Resource Agent

**Aggregates** *bm*, *hy*

**Attribute** virtual: true, feasible Regions:  $[0, 0], [50, 700]$ , cost: \*piecewise linear function\*

Figures 2 and 3 show the modeling of the two levels of hierarchy using object diagrams. In the lower side of the picture the lowest level of abstraction is represented while the upper side of the picture shows how a virtual agent can be built out of a composition of resource agents.

After the description and the illustration on the Figure, it is important to highlight that we can express constraints on production (of energy in this particular case) within instances between the association class between the *ResourceAgent* instance and the *Competency* one. Typically the *Contract* instance is part of the *Service Level Agreement (SLA)* that is expressed at *Service Realization* instance level as a composition of *Contract* instances for the realization of the service. The building of the global SLA can thus also be expressed on multiple levels and be rolled-up and drilled-down on multiple levels.

## 5 SOLVING HIERARCHICAL RESOURCE ALLOCATION

Once the resource agents are modeled adequately, we can use model abstraction to reduce the complexity of the optimization problem faced when solving cost-effective resource allocation. First, for each inner node in the hierarchy, a set of models of underlying resource agents is collected and abstracted into a single one (representing the group's capabilities) and can be handed over to the subordinate agent. Second, the resource allocation can be performed in a top-down and recursive manner. Algorithms as described in (Schiendorfer et al., 2014) come into play. We illustrate the concept with the case study and show the simplest of all techniques, finding the general feasible regions of one virtual power plant. To do so, we essentially combine all feasible intervals of the underlying agents and merge overlapping intervals. If we activate only *bm*,  $\{[100, 400]\}$  is possible, if we activate only *hy*,  $\{[50, 300]\}$  is viable. Switching both off enables the  $[0, 0]$  interval and switching both on means that the resulting interval is given by  $[150, 700]$ , hence we obtain the in-

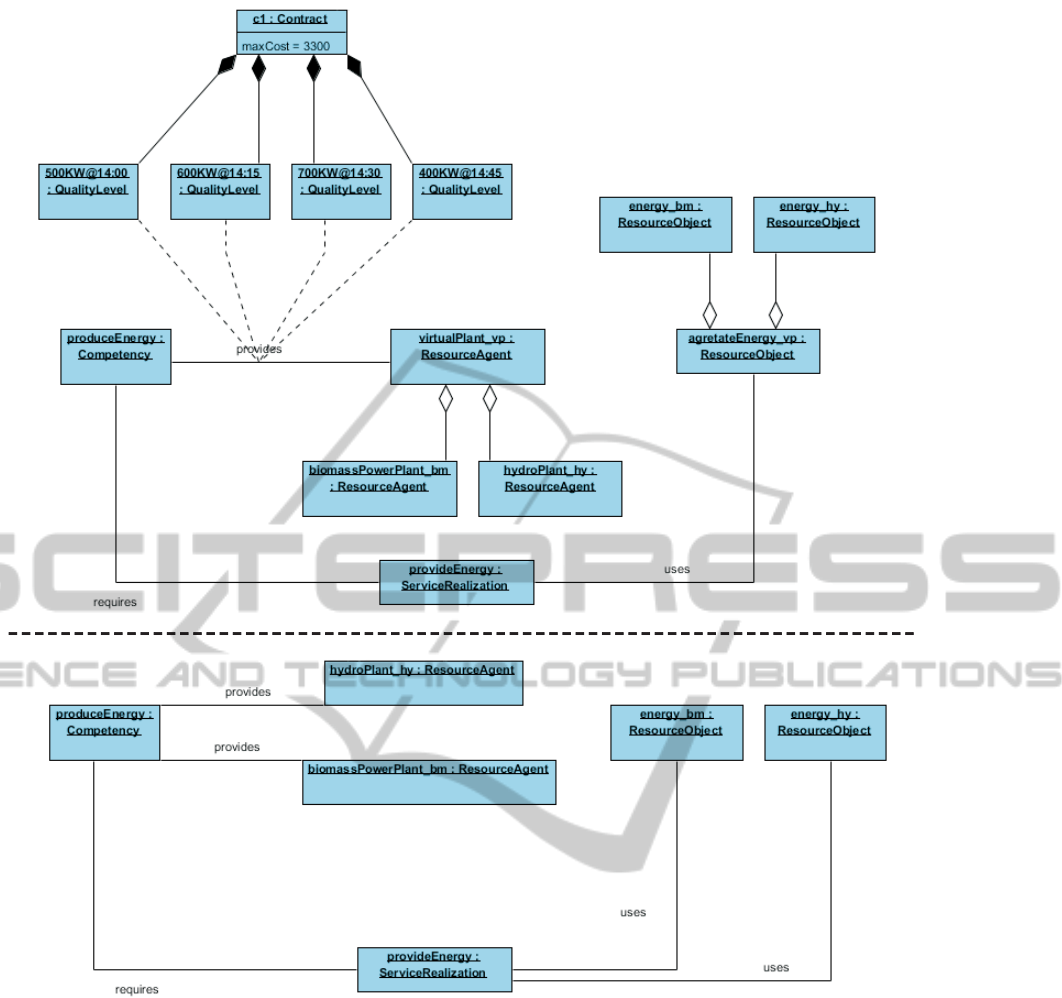


Figure 3: Resource Modeling on Various Levels of Abstraction.

tervals  $\{[0, 0], [50, 300], [100, 400], [150, 700]\}$ . However, from an outside perspective the concrete configurations do not matter (e.g., if 150 is provided by (0, 150) or (50, 100)) but only the feasible and infeasible regions. This abstracted view is achieved by merging overlapping intervals yielding  $\{[0, 0], [50, 700]\}$ .

The abstracted model does not show particulars of the underlying agents and can hence be seen as a means to perform efficient intra-organization communication as superordinate agents need only to worry about a defined interface (e.g., consisting of the available feasible regions of an abstracted *ResourceAgent* and its cost function). This cost function can be found by an approach based on sampling, i.e., repeatedly solving a combinatorial problem to build an approximation of the cost function. For our concrete example, we sketch the steps involved for mapping production to costs (input and output pairs are collected to form piecewise linear functions):

1. Solve for *minimize* costs, given that the combined production is 0
2. Solve for *minimize* costs, given that the combined production is 50
3. Solve for *minimize* costs, given that the combined production is 55
4. ...

## 6 CONCLUSION

RISE can benefit from unified modeling mechanisms particularly for environments requiring to be expressed on multiple hierarchical levels. We have, in this paper, illustrated hierarchical modeling of resources within the context of smart grids as well as the impact onto the resource allocation problem. Ex-

pressing the problem on multiple levels allows the following benefits:

- express constraints, contracts and SLA on multiple levels. Ultimately, the SLA is a composition of contracts and is located at top abstraction level, i.e. the service level;
- deal with resource allocation algorithms at multiple levels with direct impact on the other levels as depicted in (Schiendorfer et al., 2014);
- by nature, we can treat multi-organizational problems at once. Indeed, when multiple organizations are depicted at lower level, the application of optimization algorithms at upper level treats each peer uniformly for an efficient resource use and allocation at global level;
- by the associated development approach, it inherently fills the gap between resource assignment at design time and resource allocation at run time.

The work presented in the paper remains however preliminary; its main aim is to argue for the use of a unified model in a multi-hierarchical and multi-organizational context. The full instantiation and implementation of the ontology onto the smart grid case as well as on other case studies is currently under implementation.

## REFERENCES

- Cabanillas, C., García, J. M., Resinas, M., Ruiz, D., Mendling, J., and Cortés, A. R. (2013). Priority-based human resource allocation in business processes. In Basu, S., Pautasso, C., Zhang, L., and Fu, X., editors, *Service-Oriented Computing - 11th International Conference, ICSOC 2013, Berlin, Germany, December 2-5, 2013, Proceedings*, volume 8274 of *Lecture Notes in Computer Science*, pages 374–388. Springer.
- Chevaleyre, Y., Dunne, P. E., Endriss, U., Lang, J., Lemaître, M., Maudet, N., Padget, J., Phelps, S., Rodríguez-aguilar, J. A., and Sousa, P. (2006). Issues in Multiagent Resource Allocation. *Informatica*, 30(1):3 – 31.
- OMG (2011). *OMG Unified Modeling Language (OMG UML). Version 2.4*. Technical report, Object Management Group.
- Ramchurn, S. D., Vytelingum, P., Rogers, A., and Jennings, N. R. (2012). Putting the ‘Smarts’ into the Smart Grid: A Grand Challenge for Artificial Intelligence. *Commun. ACM*, 55(4):86–97.
- Schiendorfer, A., Steghöfer, J.-P., and Reif, W. (2014). Synthesis and Abstraction of Constraint Models for Hierarchical Resource Allocation Problems. In *Proc. 6<sup>th</sup> Int. Conf. Agents and Artificial Intelligence (ICAART’14), Vol. 2*, pages 15–27. SciTePress.
- Seebach, H., Nafz, F., Steghöfer, J.-P., and Reif, W. (2010). A Software Engineering Guideline for Self-Organizing Resource-Flow Systems. In *Proc. 6<sup>th</sup> Int. Conf. Self-Adaptive and Self-Organizing Systems (SASO’10)*, pages 194–203.
- Van Zandt, T. (1995). Hierarchical Computation of the Resource Allocation Problem. *European Economic Review*, 39(3-4):700–708.
- Wautelet, Y., Heng, S., and Kolp, M. (2012). A usage-based unified resource model. In *SEKE*, pages 299–304. Knowledge Systems Institute Graduate School.