

An Ontological Model for Resource Sharing in Pervasive Environments

Kimberly Garcia, Manuele Kirsch Pinheiro, Sonia Mendoza, Dominique Decouchant

▶ **To cite this version:**

Kimberly Garcia, Manuele Kirsch Pinheiro, Sonia Mendoza, Dominique Decouchant. An Ontological Model for Resource Sharing in Pervasive Environments. 2013 IEEE/WIC/ACM International Conference on Web Intelligence (WI), Nov 2013, Atlanta, GA, United States. pp.179 - 184, 10.1109/WI-IAT.2013.27 . hal-01020974

HAL Id: hal-01020974

https://paris1.hal.science/hal-01020974

Submitted on 8 Jul 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An Ontological Model for Resource Sharing in Pervasive Environments

Kimberly Garcia*, Manuele Kirsch-Pinheiro†, Sonia Mendoza*, Dominique Decouchant‡
*Department of Computer Science, CINVESTAV-IPN, México DF., México
kimberly@computacion.cs.cinvestav.mx, smendoza@cs.cinvestav.mx
†Centre de Recherche en Informatique Université Paris1-Panthéon Sorbonne, Paris, France
Manuele.Kirsch-Pinheiro@univ-paris1.fr
‡Department of Information Technologies UAM-Cuajimalpa, Mexico DF., Mexico
decouchant@correo.cua.uam.mx

***Abstract*—Since over a decade ago, Service Discovery Protocols have been developed for applications to take advantages of services shared over networks. Each protocol provides a way for applications to ask for the type of services they need. In return, applications receive information about services that could fulfill their needs. However, such protocols do not supply enough support for dynamic environments involving interaction among persons, heterogenous computer devices, applications and files. Unlike applications, people expect to receive customized information about their identity, role, social relations, and even contextual variables (e.g., location). Thus, to provide a computational support that satisfies users' requirements in an accurate way, we clinch that up to date information about the current conditions of the environment and the entities involved in it is fundamental. In this paper, we tackle the lack of a descriptive technique for resource discovery by proposing a carefully designed ontological approach.**

***Keywords*-ubiquitous collaborative environments, shared resource discovery, semantic resource description.**

I. Introduction

The main goal of ubiquitous computing [1] is to incorporate technology into the users' environment by making it so easy to use that it would become invisible to them. They will be able to focus on their tasks instead of thinking on configuring the tools the environment offers. This ubiquitous world, as dreamt by Weiser [1], is now becoming a reality with the increasing amount of devices and information available anytime anywhere. People are more habituated to receive information according to their location and preferences. Our work goes one step further by adapting such a ubiquitous environment to support collaborative work.

In a collaborative ubiquitous environment, sharing resources (e.g., devices or files) among people in a controlled way (e.g., through restrictions and access rights) is a necessity. In this case, the correct management of information can make the difference between providing a good response when requesting a resource or not. We argue that creating a proper descriptive model of this environment, which involves describing resources, the setting in which they are shared and the changes in the conditions of both the resources and the environment, constitutes the first step to cope with the previously stated limitations.

Our proposal consists of a semantic model that tackles the description question through an ontological approach. Thanks to ontologies, we are able to describe concepts representing resources and properties denoting relationships among these concepts. These relationships provide meaning to each concept involved in the modeled domain.

This paper is organized as follows. After presenting related work in section II, we illustrate in section III some scenarios the need the deployment of an architecture for resource management. In particular, we present a set of settings and introduce our case of study to validate the proposed RAMS (Resource Availability Management Service) architecture, which is described in section IV. Then, section V details the ontologies proposed for describing shared resources, and section VI illustrates the importance of these ontologies by acknowledging, in a real scenario, the amount and variety of information we are able to handle through them. In section VII we present our conclusions and future work.

II. Related Work

In this section, we analyze some outstanding works involving resource discovery. We present in section II-A, some traditional Service Discovery Protocols (SDP), and in section II-B, we discuss a couple of frameworks that have been recently developed to provide context awareness capabilities to the service discovery task.

A. Service Discovery Protocols

One of the SDP pioneers is the Service Location Protocol (SLP) [2], which was developed by IETF in order to prevent applications from having to know the specific network location of the required services. When an application needs a service, it makes a request specifying the characteristics it needs. In response, the application receives the URL of the service that fulfills its requirements. Then, it uses this URL to contact the service.

Ninja Service Discovery Service (SDS) [4] is an academic proposal, which uses predefined XML templates to describe each type of services. Ninja SDS creates representative vectors of the announced and requested services by applying

hash functions to subsets of their features. When a request is made, these types of vectors are compared. If a match is found, the requester gets a XML document containing the service information and the way to locate it.

Jini [5] is one of the most popular SDPs. Introduced by Sun Microsystems, Jini considers as a service any artifact (e.g., a coffee maker) able to be represented by a Java object. When a service is needed, Jini looks among the registered services and in case there is a service satisfying a client request, a Java object representing such a service is transferred to the client.

In terms of the way the SDPs describe resources, it was found that SLP and Jini use predefined templates to uniformly describe services and requests. Ninja SDS use XML documents, which constitute a more flexible mean for service description, since they can be enriched by adding new attributes (labels) when needed.

The way the offered services and requests are described is an improvable feature, since most of the descriptions are limited. For instance, some of the studied SDPs use predefined templates that should be filled out to describe offered or required services. This description process limits users to express their needs. Another important flaw is that SDPs consider neither the environment nor the actual conditions of the service or the requester. Consequently, the dynamism that characterize pervasive environments is not fully acknowledged. Besides, collaborative aspects related to resource management inside organizations remain ignored by SDPs. For instance, none of the studied protocols is able to consider a user's role in the organization when sharing a service. Some other drawbacks on SDPs were detected, such as the fact that most of them were mainly designed for applications asking for services, so they provide minimal or null support for human users.

B. Frameworks

In addition to the SDPs presented earlier, several context-aware computing frameworks have been proposed for service discovery in pervasive environments.

The Adaptable Intelligent Discovery of context-Aware Services (AIDAS) framework [7] offers contextual service discovery. The architecture of this framework includes a management set integrated by: a) a context manager, looking after changes in the environment; b) a profile matching engine, which compares service capabilities to the user's request; and c) a discovery manager, in charge of making the actual service match. When a user starts a discovery session, his own profile and the one of the registered devices are retrieved and assessed to create a view of accessible services according to the user context (i.e., location).

The DAIDALOS (Designing Advanced network Interfaces for Delivery and Administration of Location independent, Optimized personal Services) project [8] proposes to add a semantic layer to a traditional SDP (e.g., SLP or Jini). This semantic layer consists of: 1) an ontology, in which characteristics of services are expressed, and 2) a contextual manager, which holds pointers to contextual sources (e.g., sensors) of the entities participating in the protocol. When a user needs a service, he makes a request specifying the basic and semantic characteristics a service should present to fulfill his request. The user also provides the pointer to his context source. Then, the user's request is processed in order to obtain a set of suitable services.

III. Resource Sharing Scenarios

To provide a suited support for resource sharing in ubiquitous collaborative environments of different sizes, we built the RAMS architecture, which integrates the semantic resource description proposed in this paper. Before presenting such an architecture, it is important to understand the collaborative environments we can face, from small to large places filled with a big quantity and variety of resources. So, we consider some types of institutions in which shared resource management is vital.

Hospitals

Everyday, medical staff discusses patients' cases in meeting rooms enabled with computer equipments. These meetings can be already scheduled, such as when nurses switch shifts and give reports about patients to other nurses, or there can be urgent meetings, in which specialists make diagnostics or discuss a treatment. These meetings commonly lead to schedule tests and therapies. This situation involves the management of human (e.g., nurses and physicians), physical (e.g., rooms and medical equipment) and virtual (e.g., medical records) resources. Another evident situation of shared resource management is present in large hospital systems with multiple medical campuses, where the equipment used is often taken with patients who are transferred between hospitals. These transfers lead to lose or misplace vital equipment, which along with the time people waste locating available equipment cause a big financial impact.

Institutions as big and dynamic as hospitals arise many challenges in resource management, because of the mobility of human and physical resources. Such resources should be tracked and their context should be observed, to provide a computational support that fits this dynamic environment.

Universities and Research Centers

A multidisciplinary researcher center full of many heterogenous resources is the case of study we have chosen to illustrate the semantic description of resources. This environment was chosen because of our experience in organizations, in which a shared resource support would be of great utility. The bigger the institution is, the smaller the probability of knowing all the existing resources that are available in it is. Such, a computational support also allows to share resources among visitors, so, human relationships can be promoted. Following, a scenario illustrates a situation that the RAMS

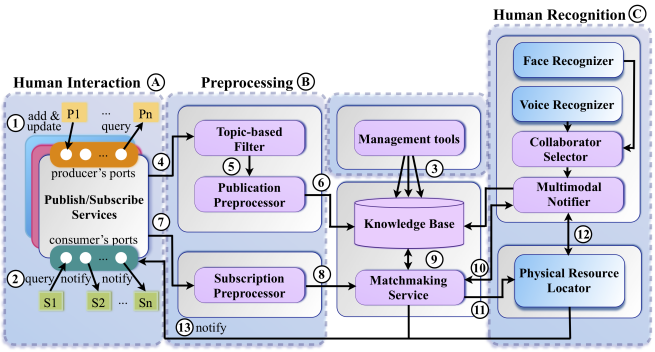


Figure 1. RAMS Architecture

architecture deployed in a university or research center is targeted to solve.

Miss Andrew requires a portable interactive whiteboard to give a conference on Ubiquitous Computing in a nearby auditorium. She has prepared some interactive material, as she likes to provide interesting presentations. However, in the auditorium where she is going to give the talk, there is no interactive whiteboard installed. As Miss Andrew considers really important to keep the audience's attention, she decides to look for a portable interactive whiteboard among her colleagues. A resource management system should help Miss Andrew by proposing her a resource for her needs and context, e.g., an interactive whiteboard near her. To give this kind of supports, it is essential to have every piece of information about the resources, collaborators and their current conditions. Section VI presents the details of how the proposed semantic description approach handles the information regarding Miss Andrew's request.

IV. The RAMS architecture

In order to accomplish our goal of providing a semantic support for describing resources in a pervasive collaborative environment, we propose a set of ontologies. They store and manage both static information (e.g., capabilities and technical characteristics, usage policies and access rights) and dynamic one (e.g., changes in a human resource's location) by creating a knowledge base. Following, we give an overview of the proposed architecture and its components, before presenting the ontologies (cf. section V).

The RAMS architecture is based on the asynchronous publish/subscribe model [9], which was chosen above other message models (e.g., message passing) since its principles can be directly applied to end-users of RAMS-based applications. Indeed, collaborators can play the roles of producers and/or consumers of events related to the state of shared resources (e.g., presence, location, and availability). However, unlike the publish/subscribe model, where producers do not know the consumers of their messages and vice versa, users of groupware applications should identify their colleagues, in order to define filters that control the scope of event production and consumption.

The RAMS architecture defines two roles that collaborators might play: 1) producers, who publish resources to share them with their colleagues and generate events to change the state of resources; and 2) consumers, who subscribe to RAMS to find out resources they need and to receive events about the state of the resources of their interest. To provide support to real collaborative environments, producers grant access rights and specify usage policies to be certain that their resources are going to be reached just by people they rely on and that the resources are going to be treated properly. This is particularly important for collaborative environments in which the availability of a resource may depend on information such as the user's role and rights inside the organization.

The components of the RAMS architecture are classified according to the type of services they provide into: human interaction (see Fig. 1-A), data preprocessing (see Fig. 1-B) and human recognition (see Fig. 1-C).

The human interaction category consists of a Broker that provides services for implementing an interaction support between collaborators and RAMS-based applications. Particularly, the Publication Service (see Fig. 1 step #1) allows collaborators playing the producer role to describe their resources in terms of technical characteristics, to define usage policies and to give access rights to colleagues they want to share their resources with. The Publication Service sends that resource-related information to the Topic-based Filter (see Fig. 1 step #4), which classifies it into the right ontology according to the type of resources that is being published (e.g., human or virtual resource). The Publication Preprocessor structures the classified information received from the Topic-based Filter (see Fig. 1 step #5) to make it comprehensible for the RAMS Ontologies and stores it into the knowledge base (see Fig. 1 step #6).

Collaborators playing the consumer role can interact with RAMS-based applications by means of the Subscription Service (see Fig. 1 step #2), which allows them to describe the type of resources (or a specific resource) they are interested in. The Subscription Preprocessor structures the resource description obtained from the Subscription Service (see Fig. 1 step #7) to make it understandable to the Matchmaking Service (see Fig. 1 step #8).

From relevant information retrieved from the RAMS Knowledge Base (see Fig. 1 step #9), the Matchmaking Service selects authorized resources, whose attributes correspond to the technical descriptions provided by collaborators when subscribing to resource information or when emitting specific requests. As a result of this matchmaking process, a set of resources that potentially satisfies the consumer's request is obtained. However, these resources cannot be considered the best match for the request, if they are not available for the consumer at a given moment. To verify

effective resource availability, the Matchmaking Service takes into account dynamic information of these resources, provided by the Multimodal Notifier (see Fig. 1 step #10) and by the Management Tools (see Fig. 1 step #3). The Multimodal Notifier is in charge of communicating the decision of the Collaborator Selector, which determines a human resource's presence and location by considering information coming from a Face Recognizer and a Voice Recognizer. These components ensure an accurate response about a collaborator's identity and location. The Management Tools allow producers to modify their availability or the one of their published resources at anytime.

When a consumer is looking for a physical resource, the set of suitable and available resources selected by the Matchmaking Service is transmitted to the Physical Resource Locator (see Fig. 1 step #11), which asks the Multimodal Notifier for the consumer's current location (see Fig. 1 step #12) in order to determine the closest resource and the path he should follow to reach it. The results produced by either the Physical Resource Locator (when looking for a physical resource) or the Matchmaking Service (when searching for a human or virtual resource) are finally delivered to the consumer (see Fig. 1 step #13).

V. The RAMS Ontologies

In order to semantically describe resources, the RAMS architecture adopts an ontological approach. Three kinds of resources are described: 1) human resources (i.e., collaborators whose position or role in the organization can be necessary to other collaborators); 2) physical resources, such as hardware (e.g., printers and projector) and rooms; and 3) virtual resources (e.g., files, software, and databases). In addition to these three ontologies (one for each kind of resources), the RAMS architecture proposes three other ontologies, which model the rest of the resource sharing environment. The first one is the context ontology, which describes the dynamism of the environment. The institution ontology specifies organizational aspects of the institution in which the RAMS architecture is deployed. Finally, the meta-ontology associates the previously mentioned ontologies. Following, section V-A introduces the proposed ontologies. Later, section V-B presents the group of object properties proposed to create relationships among individuals belonging to the same or different classes. Next, in this same section, the utility of those object properties and their restrictions are explained. Afterwards, section V-C describes a study we carried on to select the most representative characteristics of the considered resources to model them as data properties. Finally, the context ontology is explained in detail in section V-D.

A. RAMS Ontologies

Our proposal consists of a set of ontologies carefully designed to cover the description needs of different resource sharing environments. A set of ontologies are proposed instead of just one big ontology, because this segmentation makes easier the reuse, adaptation, and extension of the modeled environment. All the classes in the RAMS ontologies are disjoint, as they describe entities with unique characteristics. For reasoning purposes, none of the individuals of the RAMS system should belong to more than one class.

The Protégé tool [10] was used to create the taxonomy of the proposed ontologies [11]. This tool uses the OWL-DL language. It creates ontologies in a hierarchical class structure, in which each class extends from the Thing class. Following, we give a summary of each one of the proposed ontologies. It is also important to emphasize that these ontologies can be easily extended by adding new concepts.

Human Resource Ontology. People are the most important resource inside an organization, since without appropriate people, no organization can survive. So, collaborative environments should help them sharing their competences and characteristics to get results that are better than the simple addition of their forces. In this sense, collaborators become a resource that should be carefully managed. Indeed, people, inside an organization, may share information about themselves and the roles they play in it. To give an overview of this ontology, we describe the classes included in the first level of the ontology hierarchy:

- `HumanResource`. Each individual belonging to this class represents a collaborator involved in the resource sharing environment.
- `JobPosition`. Individuals belonging to this class represent actual job positions, which are linked to individuals representing collaborators.

Physical Resource Ontology. Physical resources are particularly important in a shared environment. They may assume different forms, from traditional hardware (e.g., projectors or computers) to the environment itself (e.g., class rooms or meeting rooms). As one form of collaboration consists in sharing this type of resources, allowing people to properly describe them is an essential feature of resource management. The basic structure of this ontology consists of two main classes:

- `Building`. An individual belonging to any subclass of this class represents an actual place in the building (e.g., classroom or meeting room).
- `Hardware`. The subclasses extending from this class group individuals representing actual shared devices (e.g., computer or projector).

Virtual Resource Ontology. This ontology allows the description of an important family of resources that can be available on a collaborative environment. Virtual resources are represented by individuals categorized in the following subclasses: `File`, `DriverPlugin`, `Database` or `Software` classes.

Institution Information Ontology. This ontology is open to group specific organization information needed to complete the model. For our case of study, which is a research center, the institution ontology describes the research fields studied in each department of our organization.
Meta-Ontology. The function of this ontology is to relate all ontologies described above and the context ontology (see section V-B) by defining object properties.

B. Object Properties

Object properties give meaning to an ontology as they relate instances of classes that otherwise are independent. Such object properties have been specified to replicate in our semantic model the interaction that individuals have in the real world. They have been classified in different groups according to their purpose. Following, we present just some of the most representatives ones:

1) *humanResourceProperty*. This group of properties relates human resources to their institution information. Such properties can be customized according to the content of the institution information ontology.
2) *locationProperty*. These properties relate individuals representing physical or human resources to individuals representing physical or virtual places.
3) *resourceRelationProperty*. This group of properties specifies the owner or responsible of a virtual or physical resource by relating individuals representing human resources to individuals representing physical or a virtual resources.
4) *constraintProperty*. Properties categorized in this group create a three - individual - relationship to define the usage restrictions a human resource should satisfy when using a virtual or physical resource.
5) *taskProperty*. These properties define who is authorized to perform a task over a virtual or physical resource.
6) *collaborativeContext*. It groups object properties integrating the organizational context model. They are presented in detail in section V-D.

Characteristics and Closure Axioms

The meaning of object properties can be more valuable if their characteristics are determined. For that reason, each object property of the proposed model was analyzed and their characteristics were determined. Following we present the most relevant characteristics considered along with an object property of the RAMS ontologies:

Inverse properties. They define relationships in both directions (i.e., from individual **a** to individual **b** and from **b** to **a**).
Functional properties. An object property with this characteristic should relate at most one individual to another. Let us consider an individual belonging to the `Hardware` class (i.e., a device): it can be located at just one place at a time, so the relationship is functional.

Transitive properties. It defines that an object property relating the individual **a** to the individual **b**, and the individual **b** to **c** is capable of relating the individual **a** to the individual **c**. The *isColleagueOf* object property has this characteristic, making possible that different individuals from the `HumanResource` class relate to each other.

Closure Axioms. These axioms are proposed to complete our ontology model. Let us consider the *isLocatedAt* object property, which relates individuals from the `VirtualResource` class to individuals from the `Computer` class. Over this relationship a universal and an existential restrictions are defined. Thus, the axiom states that a virtual resource has to be located at least in one computer and that it can only be located at a computer (and not in a human resource for instance).

C. Data Properties

It was necessary to identify the data properties (e.g., speed and resolution) capable of creating the best description for the resources that the RAMS architecture proposes as generic for any environment. Thus, a survey was designed and applied to 100 potential users. The survey included technical characteristics of 18 different hardware devices (e.g., projector, printer and display) and 17 types of virtual resources (e.g., database, multimedia player and word processor). These features were obtained from vendor specifications. Potential users chose from a set of features, the ones they considered the most relevant in a resource.

The study was conducted in two institutions: 1) a university, in which 50 undergraduate students from different majors answered the questionary; and 2) our research center, in which staff, researchers, master and PhD students answered the survey. It took us over three weeks to obtain all the answers. The surveys applied in the university were paper-based, so it was faster to retrieve peoples responses. However, as people in the research center are really busy, we sent them an invitation to answer the online survey version. It took longer but we finally retrieved their preferences. Once we had the 100 answers, the features that were selected by over 50 respondents were found and considered as the main data properties for each type of resources.

To illustrate the study, in Figure 2 the Laptop features that were considered in the survey are presented. The blue and green columns represent the number of people who selected the feature as important in the paper-based and the online version respectively. The yellow column corresponds to the total of answers in both versions.

D. Context Ontology

One of the main characteristics of our proposal is the interest in performing shared resource discovery considering the

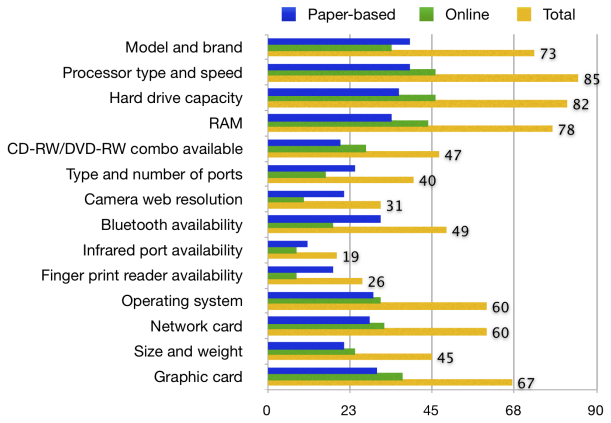


Figure 2. Laptop Data Properties Selection

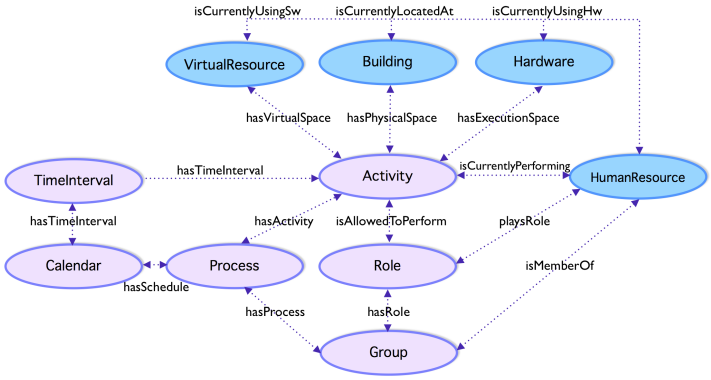


Figure 3. Organizational Context Model

environment. Thus, variables that describe this environment and its changes have to be evaluated, in order to determine the availability degree of a resource. These variables are represented in the context ontology.

Context information according to Dey [12] refers to any information that can be used to characterize the situation of an entity (a person, a place or an object), which is considered relevant to the interaction between a user and an application. Several authors ([7], [8]) have proposed to represent context information using ontologies. However, according to Najar et al., [13] most of the context representations are user-centric, remaining limited to physical aspects such as user location and device capabilities. These representations do not consider the user in his collaborative environment. Yet, other authors ([14], [15], [13]) have pointed out the importance of organizational information on the context definition. Similar to Najar et al., [13], we also consider that context representation should evolve to include not only physical aspects but also organizational ones.

Thus our context ontology considers two families of information: organizational and physical context. The first one assimilates the liveliness of a collaborative environment. While the later stands for dynamic information gathered directly from the physical environment.

Organizational Context

To model the organizational context, we consider Kirsch's work [15], which proposes an organizational context model for an awareness mechanism embedded on a groupware system supporting asynchronous work. This model considers five features to represent context: space, tool, community, time and process. The objective of this model is to provide awareness information for a set of activities (process) performed asynchronously (time) by a group of collaborators (community) using a tool (device and application) in a physical location (space). However, the RAMS objective is not as fine grained as the target of the model proposed in [15]. So, we take this model as a base and modify it to allow RAMS to provide a contextual response when resources are required for reaching a goal.

The adaptation of the original model was incorporated to the context ontology by adding the following subclasses to the `OrganizationalContext` class:

- `Process`. Each individual belonging to this class is related to a final goal (e.g., meeting) that can be reached by a collaborator or a group.
- `Activity`. An individual from this class represents an activity that can be performed in a process.
- `Group`. Each individual from this class represents a set of collaborators.
- `Role`. An individual belonging to this class represents a part a collaborator has to play when performing an action.
- `Calendar`. Individuals from this class represent the schedule of a process.
- `Time interval`. This class holds individuals representing a period of time.

Object properties were defined to relate the concepts of this model and the three types of resources managed by RAMS (see Fig. 3). Thus, an individual **h** from the `HumanResource` class belongs to a group represented by an individual **g** from the `Group` class (**h** *isMemberOf* **g**). This group **g** has a goal to reach, which is represented by an individual **p** from the `Process` class; so, **g** *hasProcess* **p**. This process **p** is associated to a schedule **c**, which is an individual from the `Calendar` class, thus **p** *hasSchedule* **c**. This calendar **c** is also linked to one or many periods of time **t**, which is an individual from the `TimeInterval` class, so **c** *hasInterval* **t**.

As the group **g** is the representation of a set of collaborators who can play specific roles, a relationship between **g** and an individual **r** from the `Role` class is created as follows **g** *hasRole* **r**. To determine the actual role of each

collaborator **h** and **r** are related (**h** `playsRole` **r**). The role **r** is associated to an individual **a** from the `Activity` class to specify the allowed activities a role can perform (**r** *isAllowedToPerform* **a**). Such activity **a** is performed inside a place **p**, which is an individual from the `Building` class (**a** *hasPhysicalSpace* **b**). As the RAMS objective is to provide resource sharing support, it is extremely important to consider the virtual and physical resources used to perform an activity. Thus, **a** *hasExecutionSpace* **hw**, where **hw** is an individual from the `Hardware` class. For an activity performed using a virtual resource, the relationship **a** *hasExecutionSpace* **sw** is defined, in which **sw** is an individual from the `VirtualResource` class. To complete the model, object properties denoting the actual usage of the resources involved in an activity that is currently taking place are created. Thus, **h** *isCurrentlyPerforming* **a**. To define the physical resource being used, the relationship **h** *isCurrentlyUsingHw* **hw** is used. Likewise, for a virtual resource, the association **h** *isCurrentlyUsingSw* **sw** is defined. Finally, a link between **h** and **p** is created as follows **h** *isCurrentlyLocatedAt* **p**.

Some of the main advantages of this model are that an activity and a role can be also a consumer, so a scheduled activity can reserve the resource it is going to need. Also, at giving roles to human resources, a collaborator could benefit from having rights over resources that he usually has limited or no access to.

Physical Context

To represent the conditions of the physical environment in which resources are shared, there were defined two subclasses belonging to the `PhysicalContext` class and data properties. The subclasses are the following:

- `Restriction`. This class holds individuals representing the current value of the usage restrictions defined by a producer. To specify a usage restriction, each individual from this class is related to data properties that define its metric, and allowed and consumed value. Thus, an individual from the `Restriction` class can be related to an individual representing a physical or a human resource by the `isAssociatedTo` object property. An individual from the `HumanResource` class is then related to an individual from the `Restriction` class by the `hasToSatisfy` object property.
- `Task`. This class represents actions collaborators can actually perform over virtual or physical resources. A human resource is related to one or some individuals from this class to denote the type of actions he can perform over a virtual or physical resource.

The modeled data properties are:

- *PhysicalLocation*. This property is in charge of holding information about the location of a resource in terms of coordinates.

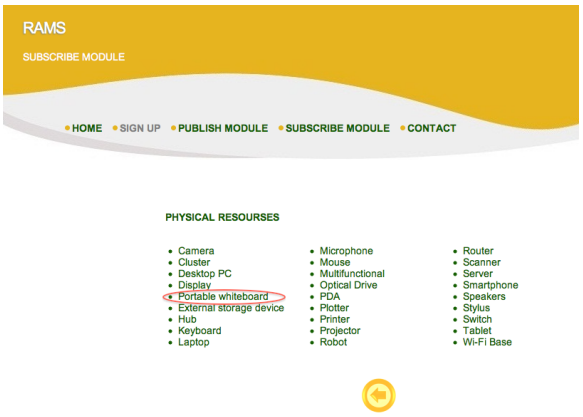


Figure 4. Laptop Data Properties Selection

- *DeviceCharacteristic*. This is a group of data properties, each one of them corresponds to the current situation of an individual from the `Hardware` class (e.g., available memory and running applications).
- *EnvironmentCondition*. This is also a group of data properties that is in charge of representing the current status of the conditions (e.g., temperature and humidity) fulfilled by an individual from the `Building` class.

VI. Model Implementation and Validation

The publish and subscribe modules were constructed as a way to provide easy interaction between collaborators and the information managed by an application built over the RAMS architecture. These modules are Web applications that use OWL-API to handle the created ontologies.

To illustrate all the information that can be managed by the proposed set of ontologies, which becomes a knowledge base once it is populated, let us consider Miss Andrew's case of study (from section III). She is interested in using a UBoard portable interactive whiteboard. She prefers this device because she is familiar with it and the required software is already installed in her laptop. By using the subscribe module, Miss Andrew expresses that she needs a portable interactive whiteboard (see Fig. 4). She also makes known that she prefers a UBoard device, but she is open to different brands. The information that can be retrieved from the knowledge base includes static and dynamic information from the already published resources. From the static information, it can be known that the following devices are being shared:

1) A UBoard belonging to Mr. Fowler;
2) An ONfinityCM2 whiteboard owned by Miss Park;
3) A public UBoard guarded by the academic secretary;
4) An eBeam whiteboard property of Mr. Thomas.

Regarding permissions and access rights, it can be known that Miss Andrew is able to use the four devices, but she has to satisfy the following restrictions:

1) Mr. Fowler is willing to share his UBoard with Miss Andrew when available;
2) Miss Park shares her ONfinity CM2 with Miss Andrew from Wednesday to Friday, but just after 2pm;
3) The public UBoard can be used by any collaborator at any moment;
4) Mr. Thomas is not able to share his whiteboard with Miss Andrew on Tuesday because he gives a lecture in another university; therefore, he is not present to give access to his whiteboard.

From observing the conditions of the collaborators and the environment when the request is made, the following information is kept in the knowledge base:

1) Mr. Fowler is inside his office, no one is using the UBoard and he is available;
2) Miss Parks is in the lunchroom. So, she can not give access to the whiteboard that is inside her office;
3) The public UBoard is being used by another professor;
4) Mr. Thomas is working with a colleague in a contiguous office, but Miss. Dubois, a PhD student, is inside Mr. Thomas' office, she is available and allowed to give access to her advisor's resources.

This information is essential to provide resource discovery at the RAMS architecture level objective, which involves not just a type of resources and its features, but also the collaborator's current conditions and their environment.

VII. Conclusion and Future Work

The deployment of the RAMS architecture in any type of organizations gives advantages to both, people working there and the organization itself. Collaborators sharing resources have full control on their resources by freely restricting the tasks and people allowed to use them. So, the safety of expensive resources is not compromised. The creation of a pervasive environment prevents collaborators from wasting time and energy looking for available resources, which also saves money to organizations.

To provide collaborators with all these benefits, an adequate way of describing resources and the environment in which they are shared is one of the main necessities. In this paper we proposed a semantic approach consisting of a customizable model expressed as a set of ontologies that suits many different type of organizations, from small ones to large and complex organizational structures.

The proposed ontological approach for describing shared resources was designed in a generic fashion to be easily implemented in any type and size of organizations. So, the current state of our research leads us to deploy this ontological approach in different environments to validate its versatility and completeness.

References

[1] M. Weiser, *The Computer for the 21st Century*, SIGMOBILE Mob. Comput. Commun. Rev., ACM, pp., 3-11, New York, NY, USA, 1999.

[2] W. Zhao and H. Schulzrinne, *Enhancing Service Location Protocol for Efficiency, Scalability and Advanced Discovery*, Journal of Systems and Software, vol. 75, num. 1-2, pp. 193-204, February, 2005.

[3] UPnP Forum *UPnP Device Architecture 1.0*, Contributing Members of the UPnP Forum, October, 2008.

[4] S. Herborn, Y. Lopez and A. Seneviratne, *A Distributed Scheme for Autonomous Service Composition*, In Proceedings of the First ACM International Workshop on Multimedia Service Composition, pp. 21-30, Singapore, November, 2005.

[5] R. Zhao, J. Y. Zhi, G. D. Liu; , *Research and implementation of distributed testing system based on Jini technology*, Antennas, Propagation & EM Theory (ISAPE), 2012 10th International Symposium on, pp.1260-1263, October, 2012.

[6] K. White, *Apple Training Series Mac OS x Support Essentials*, Peachpit Press, Berkeley, CA ,USA, 2007.

[7] A. Toninelli, A. Corradi and R. Montanari, *Semanitc-based Discovery to Support Mobile Context-aware Service Access*, Computer Communications, vol.31-5, pp. 935-949, March, 2008.

[8] V. Suraci, S. Mignanti and A. Aiuto, *A Context-aware Semantic Service Discovery*, In Proceedings of the 16th IST Mobile and Wireless Communication Summit, IEEE Computer Society, Budapest, Hungary, July, 2007.

[9] G. Muehl, L. Fiege, and P. Pietzuch, *Distributed Event-Based Systems*, First Edition, Springer-Verlag, Heidelberg, Germany, 2010.

[10] M. Horridge et. al., *A Practical Guide To Building OWL Ontologies Using Protege 4 and CO-ODE Tools*, Edition 1.3, University of Manchester, March, 2011.

[11] A. Dean; J. Hendler, *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*, Elsevier, April, 2008.

[12] A.K. Dey, *Understanding and Using Context*, Personal and Ubiquitous Computing, vol. 5, n.1., pp. 4-1, 2001.

[13] S. Najar, O. Saidani, M. Kirsch-Pinheiro, C. Souveyet and S. Nurcan, *Semantic Representation of Context Models: a Framework for Analyzing and Understanding*, In Proceedings of the 1st Workshop on Context, Information and Ontologies, CIAO' 09, ACM, Heraklion, Greece, 2009.

[14] P. Dourish, *Seeking a Foundation for Context-aware Computing*, Human Computer Interaction, vol. 13, num. 2-4, pp. 229-241, Hillsdale, 2001.

[15] M. Kirsch-Pinheiro, J. Gensel and H. Martin, Internation Workshop on Groupware, CRIWG 04, vol. 3198 of Lecture Notes in Computer Science, Springer, pp. 339-348, San Carlos, Costa Rica, 2004.