

Fujitsu Laboratories LTD. & Ricoh Company, Ltd.

OKAR Working Group

2005-03-31

# Ontology for Knowledge Activity Resources (OKAR)

## Guide

Draft 2005-03-31

---

dc:title = Ontology for Knowledge Activity Resources (OKAR) Guide Draft 2005-03-31

dc:date = 2005-03-31

dc:creator = OKAR Working Group

dc:right = Copyright © 2004-2005, FUJITSU LABORATORIES LTD. & RICOH COMPANY, LTD

## Contents

<b>1. Introduction</b> .....	1
<b>1.1. Document Status</b> .....	1
<b>1.2. OKAR Necessity</b> .....	1
<b>1.3. OKAR Use Cases</b> .....	2
<b>2. Design Policy</b> .....	4
<b>3. Specification Overview</b> .....	5
<b>3.1. Namespace</b> .....	5
<b>3.2. Basic Class Definition Overview</b> .....	5
<b>3.2.1. Basic Class Hierarchy</b> .....	5
<b>3.2.2. Agent Class</b> .....	6
<b>3.2.3. Role Class</b> .....	6
<b>3.2.3. Event Class</b> .....	8
<b>3.2.4. Artifact Class</b> .....	8
<b>3.2.5. Auxiliary Classes</b> .....	9
<b>3.3. Basic Property Definition Overview</b> .....	9
<b>3.3.1. Imported Property and Extension Definition</b> .....	9
<b>3.3.2. Unique OKAR Properties</b> .....	11
<b>3.4. Instance Description</b> .....	12
<b>3.4.1. General Description</b> .....	12
<b>3.4.2. Organization Hierachy Description</b> .....	13
<b>3.4.3. Personnel Change Description</b> .....	14
<b>4. Extension Overview</b> .....	17
<b>4.1. Extension Policy and Method</b> .....	17
<b>4.2. Extension Examples</b> .....	17
<b>4.2.1. Class Extension</b> .....	17
<b>4.2.2. Property Extension</b> .....	18
<b>5. Examples</b> .....	20
<b>5.1. OKAR Working Group</b> .....	20
<b>5.2. Change of Organization</b> .....	21
<b>5.3. Work Experiences</b> .....	22
<b>5.4. Required Knowledge</b> .....	24
<b>6. Homepage &amp; Contact Address</b> .....	26
<b>7. Change Log</b> .....	26
<b>Appendix: OKAR OWL Definition (N3 Format)</b> .....	i

## 1. Introduction

This document gives an overview of Ontology for Knowledge Activity Resources (OKAR) that is an output of the joint research working group about knowledge activity information management by Fujitsu Laboratories Ltd. (Fujitsu Laboratories) and Ricoh Company, Ltd. (Ricoh).

Although OKAR is a product of the joint research efforts of Fujitsu Laboratories and Ricoh, a third party can also use it for commercial use free of charge. Currently, the group is trying to gain supporters for OKAR and would thus greatly appreciate any feedback related to this document.

### 1.1. Document Status

This document will have the following status when it is made public on February 1st, 2005.

- † It is regarded as a working draft that can be reviewed by working group members and other interested parties.
- † It can be updated, replaced, and even scraped by another document.
- † Contact addresses in Chapter 6 to access updates and to send comments related to this overview draft.

### 1.2. OKAR Necessity

The importance of knowledge inside enterprises is greater than ever. It would be ideal if all the knowledge is documented, stored, and shared in a knowledge-base. However, sharing personal knowledge in enterprises has been a great difficulty as shown in the Gartner's report ("Knowledge Worker Investment Paradox," 2002):

- † *Employees get 50%-75% of their relevant information directly from other people,*
- † *More than 80% of enterprise's digitized information reside in individual hard drives and personal files,*
- † *Individuals hold the key to the knowledge economy and most of it is lost when they leave the enterprise.*

To manage this personal knowledge, it is not sufficient to share knowledge only in the information resources that have been already documented. It needs a mechanism to manage and update working information about what kind of information employees and groups are creating, obtaining, and processing in their knowledge activities.

The relation among people and information (knowledge) takes many shapes in work activities. The simplest is "creator" relation between a document and its author.

Another relation is the one between reference material used in a meeting and its participants. Acquaintance relations between two people who work together are also important to realize KnowWho. All these relations constitute work activity knowledge.

With the spread of the Internet and broadband connectivity, not only various systems but also information equipment are interconnected and information flows among them. In the office, networks of computers, printers, and enterprise resource planning packages (ERPs) improve work efficiency. Workers utilize the networked information equipment and systems to present documents and to communicate. It is an environment where workers connect with each other and with information. This information also constitutes work activity knowledge.

To make good use of this type of work activity knowledge from various resources, we cannot rely on specific information equipment, system, or organization. What we require is a common format (ontology) to encompass heterogeneous activities. If the ontology could be based on various types of enterprise information equipment and systems, we could output and store work activity information and automatically manage personal information from heterogeneous sources as internal enterprise knowledge.

Fujitsu Laboratories and Ricoh have jointly developed an ontology that describes information in a variety of intellectual work activities. This ontology is tentatively called “*Ontology for Knowledge Activity Resources (OKAR)*.”

OKAR pays attention to the people and the things involved with work activities. It describes the basic information classes and their properties and relations. Put concretely, OKAR can describe the people and organizations that make up an enterprise, the documents produced at work, and the event information such as for a work meeting. OKAR can also describe the relationships between all these elements.

OKAR is defined by “*Web Ontology Language (OWL)*.” In the future, we expect the semantic web to expand and metadata based on OWL to increase. These developments will enable the easy exchange of information.

### **1.3. OKAR Use Cases**

OKAR is designed to be used in various knowledge management applications. For example, it can be used to manage both skill information about workers and projects and event information related to the projects.

At the 2nd International Semantic Web Conference (ISWC) 2004 Exhibition<sup>1</sup>, Fujitsu Laboratories and Ricoh present demonstrations as sample use cases of OKAR.

Fujitsu Laboratories has developed a meeting information management system, which integrates various information equipment, based on “*Task Computing*” technology<sup>2</sup>. This system manages meeting participants based on RFID tags<sup>3</sup> and integrates various information equipment in a meeting room, which enable presentations to be made easily. At the same time, activity information such as who is giving a presentation to whom using what reference material and who is participating in what meeting is automatically recorded in OKAR format. This system also automatically analyzes the recorded information and enables an advanced search that is called “*KnowWho*”<sup>4</sup>. For example, KnowWho can search for a person who has a certain skill required in a certain job, and it can also find social network information such as who is working with whom.

Ricoh has developed a prototype knowledge resources search engine system that links with “*Document Highway*”<sup>5</sup>. This system can search for and extract knowledge from a variety of resources like people, organizations and projects that is derived from human relations information in the projects for internal document management systems and the “*Friend of a Friend*”<sup>6</sup> (FOAF). Combined with XACML<sup>7</sup>, this system can also provide vital control of information access across enterprise and organization borders.

---

<sup>1</sup> ISWC2004: <http://iswc2004.semanticweb.org/>

<sup>2</sup> Task Computing: A framework proposal by Fujitsu Laboratories, Fujitsu Laboratories of America, and Maryland University to link various types of information equipment based on semantic web technology. <http://taskcomputing.org>

<sup>3</sup> Radio Frequency Identification: A small wireless IC chip attached to a person or a thing for identification recognition.

<sup>4</sup> KnowWho: Searching for person-centric work information like who possesses what knowledge. Efficiently finding people who possess the knowledge required for a task is essential to strengthen the competitiveness of an enterprise. This field of internal information sharing and knowledge management is attracting a lot of attention.

<sup>5</sup> Document Highway: A platform proposed by Ricoh that links multi-functional digital machines, information equipment like printers, and applications over a network. The platform promotes the smooth flow and management of office information. <http://www.ricoh.co.jp/src/en/highway/>

<sup>6</sup> FOAF: <http://www.foaf-project.org/>

<sup>7</sup> XACML: <http://www.oasis-open.org/committees/xacml/>

## 2. Design Policy

The objective of OKAR is to describe the general intellectual work activity in an office. The main description targets of activity are people and things. Basic resource classes of OKAR are **Agent** (perform activity), **Artifact** (target of activity, result of activity), and **Event** (activity performed by agent, action itself).

OKAR also introduces a resource class **Role**, which describes the diversity of an **Agent**. For example, a person may belong to multiple organizations and have a different role in each. A role may also change over time. By defining **Role** as a resource, OKAR can accurately describe the diversity of **Agent** and the role change over time.

The initial version of OKAR does not comprehensively define all the resources that appear in an office environment. It only defines basic core classes. The working group used Fujitsu Laboratories and Ricoh as models to design the basic classes. The group selected common resources that appeared between the two companies and resources thought necessary to exchange information between the two companies. It also describes its own activity as a real example of intellectual work activity. Refer to Chapter 3 for details about the basic classes.

Depending on needs, a user can use extension definitions to describe OKAR basic classes in more detail. Refer to Chapter 4 for details about extension definitions.

The working group made use of vocabulary in other standards to define resource properties and relationships. OKAR classes and properties were designed only when the vocabulary was found insufficient. In this way, we have given consideration to interoperability with legacy applications.

### 3. Specification Overview

This document is an overview draft and the OKAR specification description in this chapter is subject to change without prior notice. For details about spec updates after the publication of this draft, access the OKAR Homepage shown in Chapter 6.

#### 3.1. Namespace

Below is the OKAR namespace.

```
http://www.labs.fujitsu.com/jp/techinfo/okar/0.91#
```

OKAR users must first declare the above namespace at the beginning of the “*Resource Description Framework (RDF)*” metadata. OKAR also uses vCard, iCalendar, and Dublin Core vocabularies, so the RDF/XML header generally looks like the following.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:okar="http://www.labs.fujitsu.com/jp/techinfo/okar/0.91#"
  xmlns:vcard="http://www.w3.org/2001/vcard-rdf/3.0#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:ical="http://www.w3.org/2002/12/cal/ical#"
>
...
```

#### 3.2. Basic Class Definition Overview

OKAR defines four core classes and seven derivative subclasses (owl:subClassOf), which together make up the 11 basic classes of OKAR.

##### 3.2.1. Basic Class Hierarchy

**Agent**, **Role**, **Event**, and **Artifact** make up the four basic classes. **Person**, **Organization**, **Equipment**, and **Software** are the subclasses of **Agent**, **Action** and **GroupEvent** are the subclasses of **Event**, and **Document** is the subclass of **Artifact**. OKAR also defines two auxiliary classes, which are **Location** and **PersonName**.

Figure 1 shows the basic class hierarchy along with the auxiliary classes. The oval-shaped labels show each class and the dotted-line arrows show the subclass relationship (owl:subClassOf).

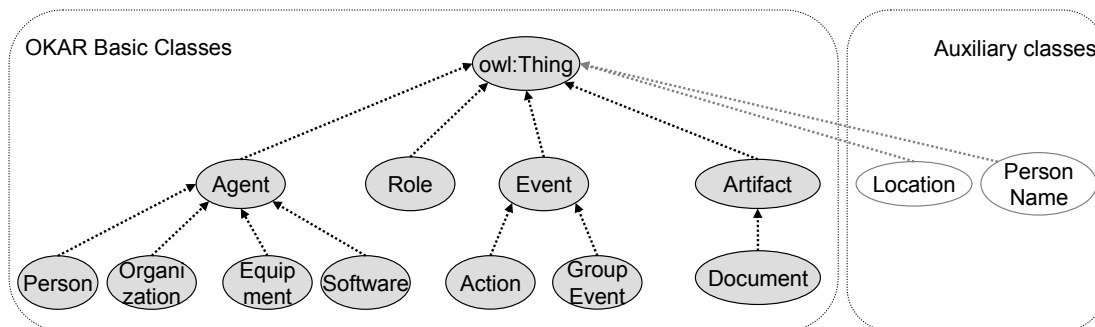


Figure 1. Basic Class Hierarchy

### 3.2.2. Agent Class

The **Agent** class describes the subject of an event or a thing that has an identity like a knowledgeable person. Mainly, this class includes **Person** and **Organization**, which are defined as subclasses. It also includes **Equipment** and **Software**, which are inorganic things.

The **Agent** class mainly uses vCard vocabulary to hold its properties. A general **Agent** instance is described below (using the **Person** subclass).

```

<okar:Person rdf:about="#person:TaroYamada">
  <vcard:FN>Taro Yamada</vcard:FN>
  <vcard:N rdf:parseType="Resource">
    <vcard:Family>Yamada</vcard:Family>
    <vcard:Given>Taro</vcard:Given>
  </vcard:N>
</okar:Person>
  
```

The **Person** class is used to describe a person, the **Organization** class is used to describe a company organization (including a temporary group), the **Equipment** class is used to describe a physical thing (computer, information equipment, etc.), and the **Software** class is used to describe a non-physical thing (application, system, etc.).

### 3.2.3. Role Class

The most versatile class in OKAR is **Role** class because it can connect the **Agent** class and other (non **Role**) basic classes. The class is used whenever relationships between the two classes exist.

In general, it is assumed that an **Agent** instance has more than one **Role**. For example, a **Person** can be a member of multiple groups and **Software** can have multiple functions. If each role were identical, there would be no problem. However, the nature of a role varies depending on its relationship with other resources. The **Role** class is used to

describe an **Agent** who has multiple roles that vary depending on the relationship with a resource.

Figure 2 shows a person (Taro Yamada) who belongs two organizations (A-Division and B-Division). His role in A-Division is Director and his role in B-Division is Senior Researcher.

If the **Person** and **Organization** relationship were expressed as one simple property (such as, `vcard:ORG`), then we would not be able to express the two office positions of the **Agent**. To solve this problem we use two **Roles** (linked with **Organization** by `okar:member`) each of which derives from an organization and has its own `vcard:TITLE`. In other words, one **Role** instance corresponds to one business card.

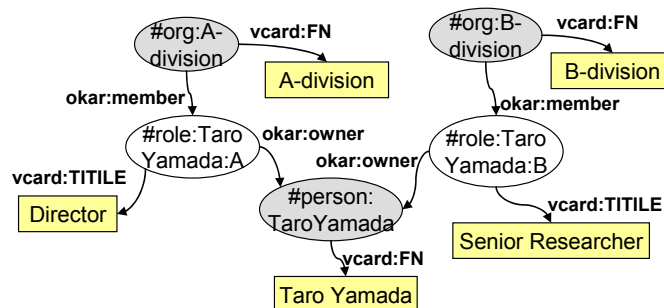


Figure 2. Agent with Different Positions in Two Organizations

```
<okar:Role rdf:about="#role:TaroYamada:A">
  <okar:owner rdf:resource="#person:TaroYamada"/>
  <vcard:TITLE>Director</vcard:TITLE>
</okar:Person>
```

The **Role** class, like the **Agent** class, can use vCard vocabulary to hold its properties. At some point, however, a conflict may occur between the **Role** class content and the **Agent** class content (linked by `okar:owner`). For example, if the **Agent** instance for Taro Yamada contained “*Taro*” as the `vcard:NICKNAME` and the **Role** instance for C-Division contained “*Yamachan*” as the `vcard:NICKNAME`, then a conflict would occur.

OKAR does not have any rule against this type of conflict, so the problem needs to be resolved at the application level. Normally, we recommend a rule in which the **Role** description content overwrites the **Agent** description content.

In addition, a **Role** instance description policy (update timing, instance change) will depend on the application used. For example, if Taro Yamada were promoted, it would be okay to describe the change in the `vcard:TITLE` (which is one **Role** attribute value) as a separate **Role** instance. In this case, the two **Roles** will describe the role changes of the **Agent** (the same as a resume for a person) by giving each **Role** instance a period of validity and ordering each instance according to time.

### 3.2.3. Event Class

The **Event** class describes who is doing what at a certain time in a certain place. The two subclasses include **Action**, which describes an action of an **Agent**, and **GroupEvent**, which describes cooperation among multiple **Agents**.

The **Event** class mainly uses iCalendar vocabulary to hold its properties. A general **Event** instance is described below (using the **GroupEvent** subclass).

```
<okar:GroupEvent rdf:about="#meeting:NextProductMeeting">
  <ical:summary>Next Product Meeting</ical:summary>
  <ical:dtstart rdf:parseType="Resource">
    <ical:dateTime>2004-10-12T15:00:00</ical:dateTime>
  </ical:dtstart>
  <ical:dtend rdf:parseType="Resource">
    <ical:dateTime>2004-10-12T17:00:00</ical:dateTime>
  </ical:dtend>
  <okar:hasLocation rdf:resource="#loc:Room-B"/>
  <ical:attendee rdf:resource="#role:TaroYamada:A"/>
  <ical:attendee rdf:resource="#role:YumikoMoriyama"/>
  <ical:attendee rdf:resource="#role:KenjiKitamura"/>
</okar:GroupEvent>
```

The example above describes a “*Next Product Meeting*” event with three participants. The event details include when, where, who, and what. As explained in section 3.3 Basic Property Definition Overview, the values taken for `okar:actor` and `ical:attendee` belong to the **Role** class. In this example, Taro Yamada attends the meeting as “*#role:TaroYamada:A*” (belonging to A-Division).

### 3.2.4. Artifact Class

The **Artifact** class describes a result produced by an **Agent** that is the subject of some action. The subclass is **Document**, which uses symbols (text or other format) to describe the thoughts and orders of an **Agent**.

The **Artifact** class mainly uses Dublin Core vocabulary to hold its properties. A general **Artifact** instance is described below (using the **Document** subclass).

```
<okar:Document rdf:about="#doc:NextProductTarget">
  <dc:title>Target for Next Product</dc:title>
  <dc:description>
    Here is a summary of the next period product target. There are four
    big targets...
  </dc:description>
  <dc:creator rdf:resource="#role:TaroYamada:A"/>
  <dc:date>2004-10-11</dc:date>
</okar:Document>
```

In the example above, Taro Yamada is the creator and the document title is “*Target for Next Product*”. As explained in section 3.3 Basic Property Definition Overview, the value taken for `dc:creator` belongs to the **Role** class. In this example, Taro Yamada creates the document as “*#role:TaroYamada:A*” (belonging to A-Division).

### 3.2.5. Auxiliary Classes

In addition to the basic classes, OKAR defines **PersonName** and **Location** auxiliary classes. **PersonName** describes the structured name of a person divided into a family name and a given name. **Location** describes the site of occurrence of an event for the **Event** class and the address, which is shown by `vcard:ADR`.

The reason for specifying **PersonName** and **Location** as auxiliary classes and not basic classes is to make it possible to import (`okar:imports`) other ontologies. For example, when it becomes possible to import the vCard ontology, **PersonName** will be replaced by vCard vocabularies. Currently there is no vocabulary in any ontologies that can be used for **Location**; but if a good one appears, it will be possible to switch to that one.

## 3.3. Basic Property Definition Overview

As described in Chapter 2 Design Policy, OKAR has an approach of making use of existing ontologies as much as possible. With this approach, we provide the interoperability with legacy applications. OKAR defines its own properties only when there is no ontology defining suitable properties. When an existing ontology can be used, its properties are imported and the property domains (`rdf:domain`) and the ranges (`rdf:range`) are given extension definitions that conform to OKAR basic classes. The next section gives an overview of each property.

### 3.3.1. Imported Property and Extension Definition

In the future, it will become possible to import vCard ontology<sup>8</sup>, iCalendar ontology<sup>9</sup>, and Dublin Core ontology<sup>10</sup> into OKAR. Currently, only the OWL compatible iCalendar can be imported. However, it is possible to have extension definitions with `rdf:domain` and `rdf:range` properties that are different from the original.

The reason vCard and Dublin Core cannot be imported currently is because they are not OWL compatible. So it is only possible to define typical properties for each ontology.

OKAR defines vCard ontology properties for use in the **Agent** class and the **Role** class,

---

<sup>8</sup> RDF vCard: <http://www.w3.org/TR/2001/NOTE-vcard-rdf-20010222/>

<sup>9</sup> RDF iCalendar: <http://www.w3.org/2002/12/cal/>

<sup>10</sup> Dublin Core: <http://dublincore.org/documents/dces/>

iCalendar ontology properties for use in the **Event** class, and Dublin Core ontology properties for use in the **Artifact** class. For example, the `rdf:domain` for `vcard:FN` (Formatted Name), which is a vCard ontology property, is defined as the `owl:unionOf` class for the **Agent** class and the **Role** class. Similarly, each `rdf:domain` for a Dublin Core ontology property is defined for the **Artifact** class. An iCalendar ontology property is defined as `owl:subClassOf` in the `ical:Vevent` class for the OKAR **Event** class. In this way, the properties that can be used in the `ical:Vevent` class of the iCalendar ontology can be described as **Event** class properties in the OKAR ontology.

The range (`rdf:range`) of each property has an extension definition range in each OKAR class. Below are some typical extension definitions.

#### § vCard extension definitions (`rdf:range`)

- \* `vcard:PHOTO, vcard:LOGO -> okar:Document class`
- \* `vcard:ADR -> okar:Location class`
- \* `vcard:BDAY -> xsd:dateTime (XML schema dateTime)`
- \* `vcard:ORG -> okar:Organization class`

#### § iCalendar extension definitions (`rdf:range`)

- \* `ical:attendee, ical:organizer -> okar:Role class`
- \* `ical:attach -> okar:Artifact class`
- \* `ical:relatedTo -> xsd:anyURI (XML schema URI)11`

#### § Dublin Core extension definitions (`rdf:range`)

- \* `dc:creator, dc:publisher, dc:contributor, dcterms:audience -> okar:Role class`
- \* `dc:date -> xsd:dateTime (XML schema dateTime)`
- \* `dc:source, dc:relation -> okar:Artifact class`

The meaning interpretation of each property is based on the meaning of each respective ontology.

---

<sup>11</sup> At <http://www.w3.org/2002/12/cal/ical#>, the range is defined using `xsd:string`, so it can take an extension definition. Originally, `okar:Event` had to be defined in `rdf:range`. However, a limited extension definition in the URI is possible because `ical:relatedTo` is declared using `DatatypeProperty`.

### 3.3.2. Unique OKAR Properties

OKAR defines unique properties that are not defined in other ontologies. For example, we can use `vcard:BDAY` of `vCard` to describe the date of occurrence (birthday) in the **Agent** class and the **Role** class. However, we uniquely define the lapse date (death day) as `okar:DDAY` because it has no relevant property.

Likewise, we define a number of organization structure properties such as `okar:member` and `okar:groupMember`.

Table 1 shows the 21 properties we uniquely define in OKAR.

Table 1. Unique OKAR Property Definitions

Property Name	Meaning	Meaning	rdf:range	Comment
DDAY	Lapse date (death day)	Agent, Role	xsd:dateTime	
hasRole	Role of Agent	Agent	Role	owl:inverseOf of owner
owner	Owner of Role	Role	Agent	HasRole same as above
member	Organization member	Organization	Role (Person)	Only Role of Person
leader	Organization leader	Organization	Role (Person)	Sub property of member
subLeader	Organization sub leader	Organization	Role (Person)	Same as above
regularMember	Regular member of organization	Organization	Role (Person)	Same as above
temporaryMember	Temporary member of organization	Organization	Role (Person)	Same as above
groupMember	Subordinate organization	Organization	Role (Organization)	Only Role of Organization
regularGroupMember	Regular subordinate organization	Organization	Role (Organization)	Sub property of groupMember, transitiveProperty
temporaryGroupMember	Temporary subordinate organization	Organization	Role (Organization)	Sub property of groupMember
relatedRole	Relationship between Roles	Role	Role	

purpose	Objective of Role	Role	xsd:string	
hasLocation	Event site of occurrence	Event	Location	See footnote <sup>12</sup>
actor	Main actor of Event	Event	Role	
target	Target of Event	Event	owl:Thing	
knows	Knowledgeable	Agent	owl:Thing	
mate	Acquaintance, workmate	Person	Role (Person)	Sub property of knows
roleWeight	Weight of Agent Role	Role	xsd:float	Value depends on application
roleRank	Organization rank	Role	xsd:float	Value depends on application
user	Real user	Artifact	Role	

As shown in Figure 1, the properties defined mainly describe the role relationship between **Organizations** and between **Organization** and **Person**. The objective of defining these properties is to fully describe the organization hierarchy and structure.

### 3.4. Instance Description

In this section, we introduce a few examples of OKAR instance descriptions. The main focus here is to show unique OKAR properties. For description examples of vCard, iCalendar, and Dublin Core vocabulary refer to their respective specifications.

#### 3.4.1. General Description

Figure 3 shows an RDF model description for a general OKAR instance, which is based on the example in 3.2 Basic Class Definition Overview. In this example, a meeting has three participants with one document used as a resource. Of the three participants, Taro Yamada belongs to two divisions and the other two each belong one division (the same as one of Yamada's divisions). Refer to 3.2 Basic Class Definition Overview for a description of the RDF/XML format.

---

<sup>12</sup>ical:location of <http://www.w3.org/2002/12/cal/ical#> is defined using DatatypeProperty, so the property that indicates the event occurrence location is defined separately.

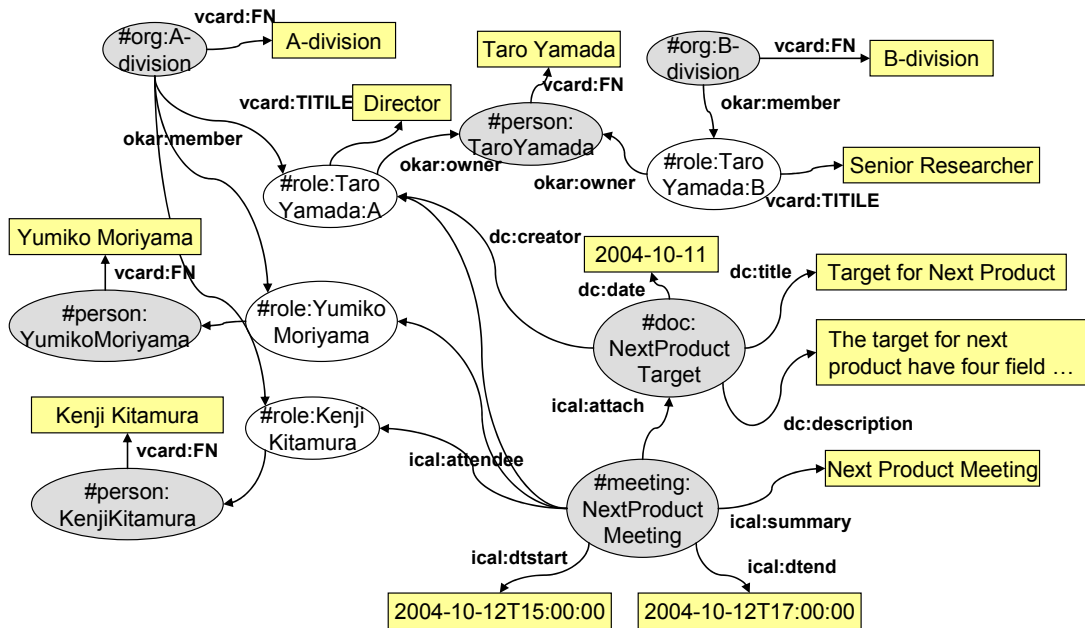


Figure 3. General OKAR Instance Description (RDF Model)

### 3.4.2. Organization Hierachy Description

As shown in Figure 4, `owl:groupMember` is used to describe the organization hierarchy. This example has A-Division, B-Division, and C-Division as subordinate organizations of AAA-Company. In addition, X-Section, Y-Section, and Z-Section are subordinate to A-Division. Note that the `rdf:domain` of `owl:groupMember` is an **Organization** instance and the `rdf:range` is a **Role** instance (of an **Organization** with `okar:hasRole`).

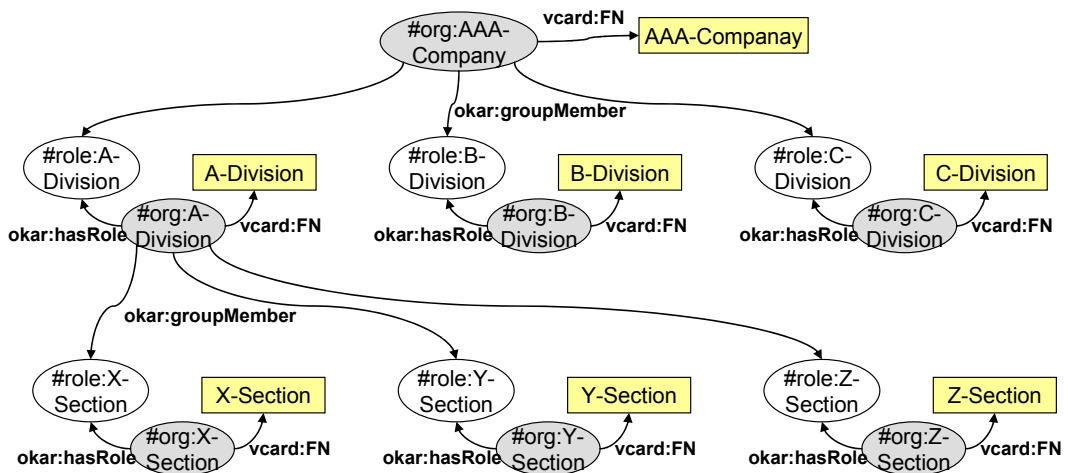


Figure 4. Organization Hierarchy Instance Description (RDF Model)

Below is an instance description in RDF/XML format.

```

<okar:Organization rdf:about="#org:AAA-Company">
  <vcard:FN>AAA Company</vcard:FN>
  <okar:groupMember rdf:resource="role:A-Division"/>
  <okar:groupMember rdf:resource="role:B-Division"/>
  <okar:groupMember rdf:resource="role:C-Division"/>
</okar:Organization>

<okar:Organization rdf:about="#org:A-Division">
  <vcard:FN>A Division</vcard:FN>
  <okar:hasRole>
    <okar:Role rdf:about="#role:A-Division"/>
  </okar:hasRole>
  <okar:groupMember rdf:resource="role:X-Section"/>
  <okar:groupMember rdf:resource="role:Y-Section"/>
  <okar:groupMember rdf:resource="role:Z-Section"/>
</okar:Organization>

<okar:Organization rdf:about="#org:X-Section">
  <vcard:FN>X Section</vcard:FN>
  <okar:hasRole>
    <okar:Role rdf:about="#role:X-Section"/>
  </okar:hasRole>
</okar:Organization>
...

```

### 3.4.3. Personnel Change Description

An organization member is described using the owl:member class. Note that the rdf:domain of owl:member is an **Organization** instance and the rdf:range is a **Role** instance (of a **Person** with okar:hasRole).

Normal affiliation information looks like the simple arrangement in Figure 3 (General OKAR Instance Description (RDF Model)). However, to show a personnel change, we describe multiple **Role** instances and each respective period of validity. The personnel change shown in Figure 5 has Taro Yamada, who was the Director of A-Division, moving to Senior Director of C-Division in 2004/10/25.

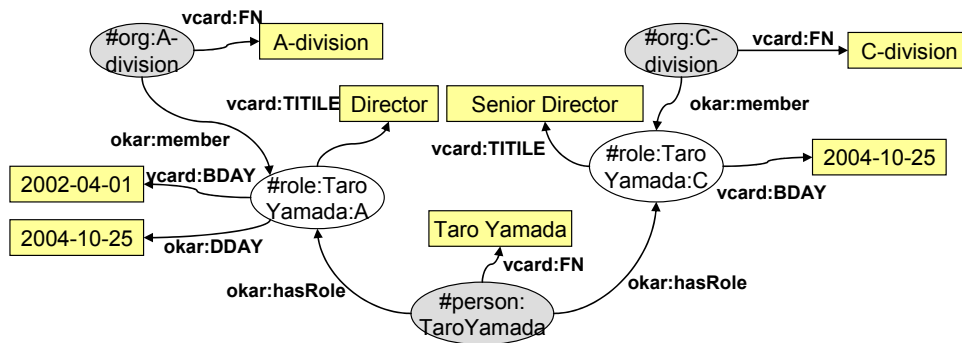


Figure 5. Personnel Change Instance Description (RDF Model)

Below is an instance description in RDF/XML format.

```

<okar:Organization rdf:about="#org:A-Division">
  <vcard:FN>A Division</vcard:FN>
  <okar:hasRole>
    <okar:Role rdf:about="#role:A-Division"/>
  </okar:hasRole>
  <okar:member rdf:resource="role:TaroYamada:A"/>
  <okar:member rdf:resource="role:YumikoMoriyama"/>
  <okar:member rdf:resource="role:KenjiKitamura"/>
  ...
</okar:Organization>

<okar:Organization rdf:about="#org:C-Division">
  <vcard:FN>C Division</vcard:FN>
  <okar:hasRole>
    <okar:Role rdf:about="#role:C-Division"/>
  </okar:hasRole>
  <okar:member rdf:resource="role:TaroYamada:C"/>
  ...
</okar:Organization>

<okar:Person rdf:about="#Person:TaroYamada">
  <vcard:FN>Taro Yamada</vcard:FN>
  <okar:hasRole>
    <okar:Role rdf:about="#role:TaroYamada:A">
      <vcard:TITLE>Director</vcard:TITLE>
      <vcard:BDAY>2002-04-01</vcard:BDAY>
      <okar:DDAY>2004-10-25</okar:DDAY>
    </okar:Role>
  </okar:hasRole>
  <okar:hasRole>
    <okar:Role rdf:about="#role:TaroYamada:C">
      <vcard:TITLE>Senior Director</vcard:TITLE>
      <vcard:BDAY>2004-10-25</vcard:BDAY>
    </okar:Role>
  </okar:hasRole>
</okar:Person>

```

For an organization change that occurs only between organizations, it is possible to describe multiple role instances as in a personnel change. If an organization is divided, we can use a new **Organization** instance to describe the division.

## 4. Extension Overview

### 4.1. Extension Policy and Method

If we want to make a more detailed description than the basic OKAR vocabulary permits, we can use separate extension definitions. It is preferable to define an extension for either a derived class of a basic OKAR class or a derived property of an OKAR property. This will give a unified meaning interpretation for basic OKAR classes and properties even when different users define multiple extensions. For extension definitions, if we want to mix descriptions by importing an ontology other than OKAR, it is preferable to map the OKAR ontology as much as possible. As for vocabulary which cannot be mapped to OKAR ontology or has no mapping, we must assume the OKAR processor would ignore the vocabulary.

When defining an extension, import the OKAR ontology and define the necessary extension.

### 4.2. Extension Examples

This section shows a number of OKAR extension definitions. The samples “&okar;” and “&local;” are character entities in the OKAR namespace and the ontology namespace of the extension definition respectively. Below is a description example.

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
<!ENTITY okar "http://www.labs.fujitsu.com/jp/techinfo/okar/0.91#">
<!ENTITY local "http://somewhere/LocalOKAR#">
]>
<rdf:RDF xmlns:okar="&okar;" xmlns:local="&local;" ...>

  <owl:Ontology rdf:about="http://somewhere/localOKAR">
    <!-- OKAR ontology import -->
    <owl:imports
      rdf:resource="http://www.labs.fujitsu.com/jp/techinfo/okar/0.91"/>
  </owl:Ontology>
...

```

#### 4.2.1. Class Extension

As seen in Chapter 2 Design Policy, in the initial version of OKAR, only a core office activity is defined as a basic class. If we want to describe other classes, we can define the extension as a subclass of a basic class.

For example, an **Event** that has a number of cooperating **Agents** will only have a **GroupEvent** class defined. So, depending on the application, it may be preferable to describe a meeting for debate objectives and a party for social objectives. In such case, we can define the extensions meeting (`local:Meeting`) and party (`local:Party`) as

subclasses of the **GroupEvent** class. Below is an example of an extension definition.

```
<owl:Class rdf:about="&local;Meeting">
  <rdfs:comment>Class for debate objective meeting </rdfs:comment>
  <owl:subClassOf rdf:resource="&okar;groupEvent"/>
</owl:Class>

<owl:Class rdf:about="&local;Party">
  <rdfs:comment>Class for social objective party </rdfs:comment>
  <owl:subClassOf rdf:resource="&okar;groupEvent"/>
</owl:Class>
```

In addition, **Document** is currently the only subclass defined in the **Artifact** class, but it is possible to imagine a number of subclasses for **Artifact**, like a product, for example. In such case, as we did in the meeting example above, we can define a product (`local:Product`) extension as a subclass of the **Artifact** class. Below is an example of an extension definition.

```
<owl:Class rdf:about="&local;Product">
  <rdfs:comment>Class for product </rdfs:comment>
  <owl:subClassOf rdf:resource="&okar;Artifact"/>
</owl:Class>
```

#### 4.2.2. Property Extension

As seen in 3.4.3 Personnel Change Description, if we want to record all changes, we describe all members as `okar:member` for each organization no matter how long a member belongs to an organization.

For example, if we want to inquire about current organization membership using only the current OKAR vocabulary, we will have to perform the three processes below based on descriptive metadata.

1. Refer to the **Role** instance which is the range of the `okar:member`.
2. Check the `okar:DDAY` value of each **Role** instance.
3. Check if the **Role** instance is currently valid.

In such case, we can define a property extension (`local:currentMember`), which indicates a current valid organization member, as a sub property of `okar:member`. Below is an example of an extension definition.

```
<owl:ObjectProperty rdf:about="&local;currentMember">
  <rdfs:comment>Extension property that indicates a current valid
  organization member </rdfs:comment>
  <owl:subPropertyOf rdf:resource="&okar;member"/>
</owl:ObjectProperty>
```

A **Person** instance is similar to the `okar:member` example. To quickly refer to a current valid role we can define a property extension (`local:currentRole`), which indicates a current valid role, as a subclass of `okar:hasRole`. We can also define a property extension (`local:mainRole`) that indicates a current main role. Below is an example of an extension definition.

```
<owl:ObjectProperty rdf:about="&local;currentRole">
  <rdfs:comment>Extension property that indicates a current valid
  role</rdfs:comment>
  <owl:subPropertyOf rdf:resource="&okar;hasRole"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&local;mainRole">
  <rdfs:comment>Extension property that indicates a current main role
</rdfs:comment>
  <owl:subPropertyOf rdf:resource="&okar;hasRole"/>
</owl:ObjectProperty>
```

We may also want to know who gave orders for a certain role under another application. In such case, we can define a property extension (`local:appoint`), which indicates an appointment relationship, as a sub property of `okar:relatedRole`. Below is an example of an extension definition.

```
<owl:ObjectProperty rdf:about="&local;appoint">
  <rdfs:comment>Extension property that indicates an appointment
  relationship </rdfs:comment>
  <owl:subPropertyOf rdf:resource="&okar;relatedRole"/>
</owl:ObjectProperty>
```

## 5. Examples

In this chapter we introduce four examples as OKAR instance.

- \* OKAR Working Group: a general example of knowledge activities.
- \* Change of Organization: an example of Organization's Role.
- \* Work experiences of a person: an example of Person's Role.
- \* Required Knowledge for a work: an example of okar:user and determs:audience.

You can download examples from OKAR homepage shown in Chapter 6.

### 5.1. OKAR Working Group

This sample shows the real activities of OKAR WG (not full description) and is the most general example of describing the knowledge activities using OKAR. This sample includes some interesting description such as following.

- \* "OKAR Working Group" was belonging to both RICOH and Fujitsu as temporary organization unit.
- \* WG members have two Role instance: Company's Role and WG's Role.
- \* One person knows the Company's Role of his colleague who is belong to the same company. But he knows only the WG's Role of the WG member who is belong to the different company.

```
<okar:Organization rdf:about="#org:ricoh">
  <vcard:FN>RICOH Company, LTD.</vcard:FN>
  <okar:temporaryGroupMember rdf:resource="#role:okar-wg"/>
  <okar:regulerMember rdf:resource="#role:aji:ricoh"/>
  <okar:regulerMember rdf:resource="#role:hirame:ricoh"/>
  ...
</okar:Organization>
```

```
<okar:Organization rdf:about="#org:fujitsu">
  <vcard:FN>Fujitsu Limited.</vcard:FN>
  <okar:temporaryGroupMember rdf:resource="#role:okar-wg"/>
  <okar:regulerMember rdf:resource="#role:yagi:fujitsu"/>
  ...
</okar:Organization>
```

```
<okar:Organization rdf:about="#org:okar-wg">
  <vcard:FN>OKAR Working Group</vcard:FN>
  <okar:leader rdf:resource="#role:aji:1"/>
  <okar:regulerMember rdf:resource="#role:yagi:okar"/>
  <okar:regulerMember rdf:resource="#role:hirame:okar"/>
  ...
  <okar:hasRole>
    <okar:Role rdf:about="#role:okar-wg"/>
  </okar:hasRole>
</okar:Organization>
```

```

<okar:Person rdf:about="#person:aji">
  <vcard:FN>Aji Minoru</vcard:FN>
  <okar:hasRole>
    <okar:Role rdf:about="#role:aji:ricoh">
      <vcard:TITLE>Chief Scientist</vcard:TITLE>
    </okar:Role>
  </okar:hasRole>
  <okar:hasRole>
    <okar:Role rdf:about="#role:aji:okar">
      <vcard:TITLE>Group Leader</vcard:TITLE>
    </okar:Role>
  </okar:hasRole>
  <okar:mate rdf:resource="#role:hirame:ricoh"/>
  <okar:mate rdf:resource="#role:yagi:okar"/>
  ...
</okar:Person>

```

## 5.2. Change of Organization

This sample shows the change of organization and is an example of describing organization's role. This sample includes the following statements.

- \* "Language Processing Division" had been from 2002-06-21 to 2004-06-20.
- \* "Intelligent Systems Division" has been from 1995-06-21 and joined "Fujitsu Laboratories LTD." at 1998-06-21.
- \* "Language & Media Division" has been from 2004-06-21.

```

<okar:Organization rdf:about="#org:fujitsu:labs">
  <vcard:FN>Fujitsu Laboratories LTD.</vcard:FN>
  <okar:regulerGroupMember rdf:resource="#role:fujitsu:labs:ling"/>
  <okar:regulerGroupMember rdf:resource="#role:fujitsu:labs:iins"/>
  <okar:regulerGroupMember rdf:resource="#role:fujitsu:labs:media"/>
</okar:Organization>

```

```

<okar:Organization rdf:about="#org:fujitsu:labs:ling">
  <vcard:FN>Language Processing Division</vcard:FN>
  <vcard:BDAY>2002-06-21</vcard:BDAY>
  <okar:DDAY>2004-06-20</okar:DDAY>
  <okar:hasRole>
    <okar:Role rdf:about="#role:fujitsu:labs:ling">
      <okar:owner rdf:resource="#org:fujitsu:labs:ling"/>
      <vcard:BDAY>2002-06-21</vcard:BDAY>
      <okar:DDAY>2004-06-20</okar:DDAY>
    </okar:Role>
  </okar:hasRole>
</okar:Organization>

```

```

<okar:Organization rdf:about="#org:fujitsu:labs:iins">
  <vcard:FN>Intelligent Systems Division</vcard:FN>
  <vcard:BDAY>1995-06-21</vcard:BDAY>
  <okar:hasRole>
    <okar:Role rdf:about="#role:fujitsu:labs:iins">
      <okar:owner rdf:resource="#org:fujitsu:labs:iins"/>
      <vcard:BDAY>1998-06-21</vcard:BDAY>
    </okar:Role>
  </okar:hasRole>
</okar:Organization>

<okar:Organization rdf:about="#org:fujitsu:labs:media">
  <vcard:FN>Language & Media Division</vcard:FN>
  <vcard:BDAY>2004-06-21</vcard:BDAY>
  <okar:hasRole>
    <okar:Role rdf:about="#role:fujitsu:labs:media">
      <okar:owner rdf:resource="#org:fujitsu:labs:media"/>
      <vcard:BDAY>2004-06-21</vcard:BDAY>
    </okar:Role>
  </okar:hasRole>
</okar:Organization>

```

### 5.3. Work Experiences

This sample shows work experiences of a person and is an example of describing person's roles. This sample includes the following statements.

- \* Jiroh Suzuki is a member of Company A.
- \* He has a JAVA certification.
- \* He moved from System Sales Section 3 to Solution Development Center.
- \* He has experiences of Java, Linux ....

```

<okar:Person rdf:about="#person:Suzuki">
  <okar:hasRole rdf:resource="#role:Suzuki-1"/>
  <okar:hasRole rdf:resource="#role:Suzuki-2"/>
  <okar:hasRole rdf:resource="#role:Suzuki-3"/>
  <okar:hasRole rdf:resource="#role:Suzuki-4"/>
  ...
  <okar:hasRole rdf:resource="#role:Suzuki-10"/>
  <vcard:FN>Jiroh Suzuki</vcard:FN>
</okar:Person>

<okar:Role rdf:about="#role:Suzuki-1">
  <vcard:BDAY>1999-04-01</vcard:BDAY>
</okar:Role>

<okar:Role rdf:about="#role:Suzuki-2">
  <vcard:BDAY>1999-05-01</vcard:BDAY>
  <okar:DDAY>2001-04-01</okar:DDAY>
</okar:Role>

```

```

<okar:Role rdf:about="#role:Suzuki-3">
  <vcard:BDAY>2001-04-01</vcard:BDAY>
</okar:Role>

<okar:Role rdf:about="#role:Suzuki-4">
  <vcard:BDAY>1999-05-01</vcard:BDAY>
  <okar:DDAY>2000-01-01</okar:DDAY>
</okar:Role>

<okar:Role rdf:about="#role:Suzuki-10">
  <vcard:TITLE>(qualification) Sun Certified Developer for the Java
Platform</vcard:TITLE>
</okar:Role>

<okar:Organization rdf:about="#organization:Sun">
  <okar:member rdf:resource="#role:Suzuki-10"/>
  <vcard:FN>Sun Microsystems</vcard:FN>
</okar:Organization>

<okar:Organization rdf:about="#organization:Company-A">
  <okar:regulerMember rdf:resource="#role:Suzuki-1"/>
  ...
  <vcard:FN>Company A</vcard:FN>
</okar:Organization>

<okar:Organization
rdf:about="#organization:Company-A:System-Sales-Section-3">
  <okar:regulerMember rdf:resource="#role:Suzuki-2"/>
  ...
  <vcard:FN>System Sales Section 3</vcard:FN>
</okar:Organization>

<okar:Organization
rdf:about="#organization:Company-A:Solution-Development-Center">
  <okar:regulerMember rdf:resource="#role:Suzuki-3"/>
  ...
  <vcard:FN>Solution Development Center</vcard:FN>
</okar:Organization>

<okar:Organization
rdf:about="#organization:Company-A:project:A-1">
  <okar:regulerMember rdf:resource="#role:Suzuki-4"/>
  ...
  <vcard:FN>Project A-1</vcard:FN>
</okar:Organization>

<okar:Document rdf:about="#document:projectA-1">
  <dc:subject rdf:resource="#thing:website"/>
  <dc:subject rdf:resource="#thing:JSP"/>
  <dc:subject rdf:resource="#thing:Linux"/>
  <dc:subject rdf:resource="#thing:JAVA"/>
  <dc:creator rdf:resource="#role:Suzuki-4"/>
  <dc:title>Document of Project A-1</dc:title>
</okar:Document>

```

```

<owl:Thing rdf:about="#thing:JAVA"/>
<owl:Thing rdf:about="#thing:Linux"/>
<owl:Thing rdf:about="#thing:JSP"/>
<owl:Thing rdf:about="#thing:website"/>

```

## 5.4. Required Knowledge

This sample shows required knowledge of a work and is an another example of describing person's role. This sample includes the following statements.

- \* Jiroh Suzuki was a teacher of JAVA programming.
- \* He is expected to have read "An Introduction to JAVA" as the teacher.
- \* He read "Curriculum for 2000" as the teacher.
- \* He had relationship to Hiroshi Tanaka as the teacher.
- \* He attended to a briefing of in-company training as the teacher.
- \* He taught a JAVA beginner's class as the teacher.

```

<okar:Person rdf:about="#person:Suzuki">
...
  <okar:hasRole rdf:resource="#role:Suzuki-7"/>
  <vcard:FN>Jiroh Suzuki</vcard:FN>
</okar:Person>

<okar:Role rdf:about="#role:Suzuki-7">
  <okar:relatedRole rdf:resource="#role:Tanaka"/>
  <okar:DDAY>2002-04-01</okar:DDAY>
  <vcard:BDAY>2000-04-01</vcard:BDAY>
  <vcard:TITLE>JAVA teacher</vcard:TITLE>
</okar:Role>

<okar:Document rdf:about="#document:JAVA-1">
  <dcterms:audience rdf:resource="#role:Suzuki-7"/>
  <dc:title>An Introduction to JAVA</dc:title>
</okar:Document>

<okar:Document rdf:about="#document:Curriculum2000">
  <dc:creator rdf:resource="#role:Tanaka"/>
  <okar:user rdf:resource="#role:Suzuki-7"/>
  <dc:title>Curriculum for 2000</dc:title>
</okar:Document>

<okar:GroupEvent rdf:about="#groupevent:briefing">
  <ical:attendee rdf:resource="#role:Suzuki-7"/>
  <okar:actor rdf:resource="#role:Tanaka"/>
  <ical:summary>Briefing of in-company training</ical:summary>
</okar:GroupEvent>

```

```
<okar:GroupEvent rdf:about="#groupevent:training-beginner">
  <ical:attach rdf:resource="#document:JAVA-1"/>
  <okar:actor rdf:resource="#role:Suzuki-7"/>
  <ical:summary>JAVA beginner's class training</ical:summary>
</okar:GroupEvent>

<okar:Organization
rdf:about="#organization:Company-A:Personnel-Department">
  <okar:regularMember rdf:resource="#role:Tanaka"/>
  <okar:temporaryMember rdf:resource="#role:Suzuki-7"/>
  ...
  <vcard:FN>Personnel Department</vcard:FN>
</okar:Organization>

<okar:Person rdf:about="#person:Tanaka">
  <okar:hasRole rdf:resource="#role:Tanaka"/>
  <vcard:FN>Hiroshi Tanaka</vcard:FN>
</okar:Person>

<okar:Role rdf:about="#role:Tanaka"/>
```

## 6. Homepage & Contact Address

We have opened OKAR website that contains the latest specifications and examples.

- \* Fujitsu Laboratories: <http://www.labs.fujitsu.com/en/techinfo/okar/>
- \* RICOH Company, Ltd.: [http://www.ricoh.com/src/lab/lab\\_us\\_uc3.html](http://www.ricoh.com/src/lab/lab_us_uc3.html)

To improve further development, comments related to this document are welcome. If you have an interest in OKAR, contact us at the following addresses

FUJITSU LABORATORIES LTD.

IT Media Laboratories, Intelligent Systems Laboratory

E-mail: [okar-question@ml.labs.fujitsu.com](mailto:okar-question@ml.labs.fujitsu.com)

Ricoh Company, Ltd.

UC Research Center, Ubiquitous Solution Lab, Software R&D Group

E-mail: [okar-question@src.ricoh.co.jp](mailto:okar-question@src.ricoh.co.jp)

## 7. Change Log

§ From Draft 2005-02-01 to Draft 2005-03-31

- \* Section 3.1 “Namespace”: Modify OKAR namespace from 0.9 to 0.91.
- \* Section 3.3.1 “Imported Property and Extension Definition”: Add `dcterms:audience`.
- \* Section 3.3.2 “Unique OKAR Properties”: Add `okar:user`.
- \* Section 5: Delete “Future Plans” and Add “Examples.”

§ From Draft 2004-11-10 to Draft 2005-02-01

- \* Section 3.1 “Namespace”: Modify OKAR namespace from URN to URL.
- \* Section 5 “Future Plans”: Add information about OKAR Homepage.

## Appendix: OKAR OWL Definition (N3 Format)

Below is the OKAR OWL definition. To conserve space, we used the N3 format. To read the definition in RDF/XML format, visit our website.

```

@prefix okar: <http://www.labs.fujitsu.com/jp/techinfo/okar/0.91#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix ical: <http://www.w3.org/2002/12/cal/ical#> .
@prefix vcard: <http://www.w3.org/2001/vcard-rdf/3.0#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

### Ontology Header
<http://www.labs.fujitsu.com/jp/techinfo/okar/0.91>
  rdf:type owl:Ontology ;
  rdfs:label "Ontology for Knowledge Activity Resources (OKAR) vocabulary" ;
  owl:imports <http://www.w3.org/2002/12/cal/ical> .

### Class

okar:Agent rdf:type owl:Class ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty vcard:FN ;
    owl:minCardinality "1"^^xsd:nonNegativeInteger
  ] ;
  owl:disjointWith okar:Role, okar:Event, okar:Artifact, okar:Location,
    okar:PersonName .

okar:Person rdf:type owl:Class ;
  rdfs:subClassOf okar:Agent ;
  owl:disjointWith okar:Organization, okar:Equipment, okar:Software .

okar:Organization rdf:type owl:Class ;
  rdfs:subClassOf okar:Agent ;
  owl:disjointWith okar:Person, okar:Equipment, okar:Software .

okar:Equipment rdf:type owl:Class ;
  rdfs:subClassOf okar:Agent ;
  owl:disjointWith okar:Person, okar:Organization, okar:Software .

okar:Software rdf:type owl:Class ;
  rdfs:subClassOf okar:Agent ;
  owl:disjointWith okar:Person, okar:Organization, okar:Equipment .

okar:Role rdf:type owl:Class ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty okar:owner ;
    owl:cardinality "1"^^xsd:nonNegativeInteger
  ] ;
  owl:disjointWith okar:Agent, okar:Event, okar:Artifact, okar:Location,
    okar:PersonName .

```

```
okar:Event rdf:type owl:Class ;
  rdfs:subClassOf ical:Vevent ;
  owl:disjointWith okar:Agent, okar:Role, okar:Artifact, okar:Location,
    okar:PersonName .

okar:Action rdf:type owl:Class ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty okar:actor ;
    owl:maxCardinality "1"^^xsd:nonNegativeInteger
  ] ;
  rdfs:subClassOf okar:Event ;
  owl:disjointWith okar:GroupEvent .

okar:GroupEvent rdf:type owl:Class ;
  rdfs:subClassOf okar:Event ;
  owl:disjointWith okar:Action .

okar:Artifact rdf:type owl:Class ;
  owl:disjointWith okar:Agent, okar:Role, okar:Event, okar:Location,
    okar:PersonName .

okar:Document rdf:type owl:Class ;
  rdfs:subClassOf okar:Artifact .

okar:Location rdf:type owl:Class ;
  owl:disjointWith okar:Agent, okar:Role, okar:Event, okar:Artifact,
    okar:PersonName .

okar:PersonName rdf:type owl:Class ;
  owl:disjointWith okar:Agent, okar:Role, okar:Event, okar:Artifact,
    okar:Location .

### Property

# vCard Properties
vcard:FN rdf:type owl:DatatypeProperty ;
  rdfs:domain [ owl:unionOf (okar:Agent okar:Role) ] ;
  rdfs:range xsd:string .

vcard:N rdf:type owl:ObjectProperty ;
  rdfs:domain okar:Person ;
  rdfs:range okar:PersonName .

vcard:Family rdf:type owl:DatatypeProperty ;
  rdfs:domain okar:PersonName ;
  rdfs:range xsd:string .

vcard:Given rdf:type owl:DatatypeProperty ;
  rdfs:domain okar:PersonName ;
  rdfs:range xsd:string .

vcard:Other rdf:type owl:DatatypeProperty ;
  rdfs:domain okar:PersonName ;
  rdfs:range xsd:string .

vcard:Prefix rdf:type owl:DatatypeProperty ;
  rdfs:domain okar:PersonName ;
  rdfs:range xsd:string .

vcard:Suffix rdf:type owl:DatatypeProperty ;
  rdfs:domain okar:PersonName ;
  rdfs:range xsd:string .
```

```

vcard:NICKNAME rdf:type owl:DatatypeProperty ;
  rdfs:domain [ owl:unionOf (okar:Agent okar:Role) ] ;
  rdfs:range xsd:string .

vcard:PHOTO rdf:type owl:ObjectProperty ;
  rdfs:domain [ owl:unionOf (okar:Agent okar:Role) ] ;
  rdfs:range okar:Document .

vcard:LOGO rdf:type owl:ObjectProperty ;
  rdfs:domain [ owl:unionOf (okar:Agent okar:Role) ] ;
  rdfs:range okar:Document .

vcard:ADR rdf:type owl:ObjectProperty ;
  rdfs:domain [ owl:unionOf (okar:Agent okar:Role) ] ;
  rdfs:range okar:Location .

vcard:TEL rdf:type owl:ObjectProperty ;
  rdfs:domain [ owl:unionOf (okar:Agent okar:Role) ] .

vcard:EMAIL rdf:type owl:ObjectProperty ;
  rdfs:domain [ owl:unionOf (okar:Agent okar:Role) ] .

vcard:BDAY rdf:type owl:DatatypeProperty, owl:FunctionalProperty ;
  rdfs:domain [ owl:unionOf (okar:Agent okar:Role) ] ;
  rdfs:range xsd:dateTime .

vcard:ORG rdf:type owl:ObjectProperty, owl:FunctionalProperty ;
  owl:inverseOf okar:owner ;
  rdfs:domain [
    owl:intersectionOf (okar:Role [
      rdf:type owl:Restriction ;
      owl:onProperty okar:owner ;
      owl:allValuesFrom okar:Person
    ] ) ] ;
  rdfs:range okar:Organization .

vcard:TITLE rdf:type owl:DatatypeProperty, owl:FunctionalProperty ;
  rdfs:domain okar:Role ;
  rdfs:range xsd:string .

vcard:ROLE rdf:type owl:DatatypeProperty, owl:FunctionalProperty ;
  rdfs:domain okar:Role ;
  rdfs:range xsd:string .

vcard:NOTE rdf:type owl:DatatypeProperty ;
  rdfs:domain [ owl:unionOf (okar:Agent okar:Role) ] ;
  rdfs:range xsd:string .

vcard:CLASS rdf:type owl:DatatypeProperty, owl:FunctionalProperty ;
  rdfs:domain [ owl:unionOf (okar:Agent okar:Role) ] ;
  rdfs:range xsd:string .

vcard:KEY rdf:type owl:DatatypeProperty ;
  rdfs:domain [ owl:unionOf (okar:Agent okar:Role) ] ;
  rdfs:range xsd:string .

# OKAR Properties
okar:DDAY rdf:type owl:DatatypeProperty, owl:FunctionalProperty ;
  rdfs:domain [ owl:unionOf (okar:Agent okar:Role) ] ;
  rdfs:range xsd:dateTime .

okar:roleWeight rdf:type owl:DatatypeProperty, owl:FunctionalProperty ;
  rdfs:domain okar:Role ;
  rdfs:range xsd:float .

```

```
okar:roleRank rdf:type owl:DatatypeProperty, owl:FunctionalProperty ;
  rdfs:domain [
    owl:intersectionOf (okar:Role [
      rdf:type owl:Restriction ;
      owl:onProperty okar:owner ;
      owl:allValuesFrom okar:Person
    ] ) ] ;
  rdfs:range xsd:float .

okar:knows rdf:type owl:ObjectProperty ;
  rdfs:domain okar:Agent .

okar:mate rdf:type owl:ObjectProperty ;
  rdfs:subPropertyOf okar:knows ;
  rdfs:domain okar:Person ;
  rdfs:range [
    owl:intersectionOf (okar:Role [
      rdf:type owl:Restriction ;
      owl:onProperty okar:owner ;
      owl:allValuesFrom okar:Person
    ] )
  ] .

okar:hasRole rdf:type owl:ObjectProperty ;
  owl:inverseOf okar:owner ;
  rdfs:domain okar:Agent ;
  rdfs:range okar:Role .

okar:owner rdf:type owl:ObjectProperty ;
  owl:inverseOf okar:hasRole ;
  rdfs:domain okar:Role ;
  rdfs:range okar:Agent .

okar:member rdf:type owl:ObjectProperty ;
  rdfs:domain okar:Organization ;
  rdfs:range [
    owl:intersectionOf (okar:Role [
      rdf:type owl:Restriction ;
      owl:onProperty okar:owner ;
      owl:allValuesFrom okar:Person
    ] )
  ] .

okar:leader rdf:type owl:ObjectProperty ;
  rdfs:subPropertyOf okar:member .

okar:subLeader rdf:type owl:ObjectProperty ;
  rdfs:subPropertyOf okar:member .

okar:regularMember rdf:type owl:ObjectProperty ;
  rdfs:subPropertyOf okar:member .

okar:temporaryMember rdf:type owl:ObjectProperty ;
  rdfs:subPropertyOf okar:member .

okar:groupMember rdf:type owl:ObjectProperty ;
  rdfs:domain okar:Organization ;
  rdfs:range [
    owl:intersectionOf (okar:Role [
      rdf:type owl:Restriction ;
      owl:onProperty okar:owner ;
      owl:allValuesFrom okar:Organization
    ] )
  ] .
```

```

okar:regulerGroupMember rdf:type owl:ObjectProperty, owl:TransitiveProperty ;
  rdfs:subPropertyOf okar:groupMember .

okar:temporaryGroupMember rdf:type owl:ObjectProperty ;
  rdfs:subPropertyOf okar:groupMember .

okar:relatedRole rdf:type owl:ObjectProperty, owl:TransitiveProperty ;
  rdfs:domain okar:Role ;
  rdfs:range okar:Role .

okar:purpose rdf:type owl:DatatypeProperty ;
  rdfs:domain okar:Role ;
  rdfs:range xsd:string .

okar:hasLocation rdf:type owl:ObjectProperty ;
  rdfs:domain okar:Event ;
  rdfs:range okar:Location .

okar:actor rdf:type owl:ObjectProperty ;
  rdfs:domain okar:Event ;
  rdfs:range okar:Role .

okar:target rdf:type owl:ObjectProperty ;
  rdfs:domain okar:Event .

okar:user rdf:type owl:ObjectProperty ;
  rdfs:domain okar:Artifact ;
  rdfs:range okar:Role .

# iCalendar Properties
ical:relatedTo rdf:type owl:DatatypeProperty ;
  rdfs:domain okar:Event ;
  rdfs:range xsd:anyURI .

ical:attendee rdf:type owl:ObjectProperty ;
  rdfs:domain okar:GroupEvent ;
  rdfs:range okar:Role .

ical:organizer rdf:type owl:ObjectProperty ;
  rdfs:domain okar:GroupEvent ;
  rdfs:range okar:Role .

ical:attach rdf:type owl:ObjectProperty ;
  rdfs:domain okar:GroupEvent ;
  rdfs:range okar:Artifact .

# Dublin Core Properties
dc:title rdf:type owl:DatatypeProperty ;
  rdfs:domain okar:Artifact ;
  rdfs:range xsd:string .

dc:creator rdf:type owl:ObjectProperty ;
  rdfs:domain okar:Artifact ;
  rdfs:range okar:Role .

dc:subject rdf:type owl:ObjectProperty ;
  rdfs:domain okar:Artifact .

dc:description rdf:type owl:DatatypeProperty ;
  rdfs:domain okar:Artifact ;
  rdfs:range xsd:string .

dc:publisher rdf:type owl:ObjectProperty ;
  rdfs:domain okar:Artifact ;
  rdfs:range okar:Role .

```

```
dc:contributor rdf:type owl:ObjectProperty ;
  rdfs:domain okar:Artifact ;
  rdfs:range okar:Role .

dc:date rdf:type owl:DatatypeProperty ;
  rdfs:domain okar:Artifact ;
  rdfs:range xsd:dateTime .

dc:format rdf:type owl:DatatypeProperty ;
  rdfs:domain okar:Artifact ;
  rdfs:range xsd:string .

dc:source rdf:type owl:ObjectProperty ;
  rdfs:domain okar:Artifact ;
  rdfs:range okar:Artifact .

dc:relation rdf:type owl:ObjectProperty ;
  rdfs:domain okar:Artifact ;
  rdfs:range okar:Artifact .

dc:identifier rdf:type owl:DatatypeProperty ;
  rdfs:domain okar:Artifact ;
  rdfs:range xsd:anyURI .

dc:type rdf:type owl:ObjectProperty ;
  rdfs:domain okar:Artifact .

dc:language rdf:type owl:DatatypeProperty ;
  rdfs:domain okar:Artifact ;
  rdfs:range xsd:string .

dc:coverage rdf:type owl:DatatypeProperty ;
  rdfs:domain okar:Artifact ;
  rdfs:range xsd:string .

dc:rights rdf:type owl:DatatypeProperty ;
  rdfs:domain okar:Artifact ;
  rdfs:range xsd:string .

dcterms:audience rdf:type owl:ObjectProperty ;
  rdfs:domain okar:Artifact ;
  rdfs:range okar:Role .
```