The SMARTERCONTEXT Ontology and Its Application to the Smart Internet: A Smarter Commerce Case Study

Norha M. Villegas[1,2,3] and Hausi A. Müller[1]

[1] University of Victoria, Victoria, Canada
hausi@cs.uvic.ca
[2] Icesi University, Cali, Colombia
nvillega@cs.uvic.ca
[3] IBM Canada CAS Research, Toronto, Canada

Abstract. In the *smart internet* interactions must be situation-aware and smart. That is, they must be realized with awareness of, and adaptation to users' individual and collective context situations. Therefore, context management is crucial to deliver contents and services that are relevant to the user's matters of concern. This paper presents the SMARTERCONTEXT ontology, our semantic web approach to context representation and reasoning applicable to user-centric domains of the smart internet. We illustrate the application of the SMARTERCONTEXT ontology using a *personal web* case study based on IBM's *smarter commerce* initiative. This case study demonstrates how our ontology supports context representation and reasoning to improve the relevance of retailer offers with respect to shopper situations. Our ontology is the core of the SMARTERCONTEXT infrastructure, our context management solution that exploits user web interactions as sources of meaningful personal context information, and empowers users to control context gathering and provisioning.

Keywords: dynamic context, smart internet, personal web, context representation, semantic web, ontologies, context reasoning, smarter commerce.

1 Introduction

In the smart internet, contents and services are discovered, aggregated and delivered dynamically, interactively, fully or semi-automatically in response to evolving user concerns, and under heterogeneous system infrastructures [1]. Therefore, the realization of the smart internet is highly dependent on its capabilities to understand the situation of users, individually and collectively, and the situation of services with respect to the matters of concern (mocs) of the users for which they are intended. Moreover, as mocs continuously evolve (e.g., the user's location, agenda, or shopping list change over time), context representation and reasoning mechanisms must be flexible enough to support, at runtime, the modeling of new context types and changes in inference rules.

The smart internet's three main principles are defined as follows: (i) a user-centric model for instinctive interactions, (ii) sessions for users and their mocs, and (iii) collective and collaborative web interactions [1]. These principles pose many different

M. Chignell et al. (Eds.): The Personal Web, LNCS 7855, pp. 151–184, 2013.

technological challenges. Among these challenges, context management (i.e., context representation, reasoning, gathering, provisioning, and the adaptation of context models and reasoners at runtime) constitute a major research problem [2,3]. First, user-centric models for instinctive interactions must include the relevant characteristics of context entities that describe the situation of users and services. Second, personal mocs must be explicitly modeled and managed as evolving context facts across sessions. Finally, collective and collaborative web interactions require the identification of not only individual but also social and activity context [2], to manage the satisfaction of individual mocs taking into account the social context sphere within which users interact.

To tackle the context awareness challenges posed by the smart internet, we developed the SMARTERCONTEXT context manager. Our solution provides an effective mechanism to model user mocs in the form of context facts. Most importantly, it tracks changes in their states to support smart internet applications in the delivery of personalized services and contents to users. This paper presents the SMARTERCONTEXT ontology which is the core of context representation and reasoning in our solution. Our ontology supports the specification of personal context information using context models that are in the form of *linked data* [4]. We designed the SMARTERCONTEXT ontology in such a way that it can be extended by either creating further layers in its hierarchical structure, or integrating existing domain-specific semantic web vocabularies into its layers. To the best of our knowledge, our ontology is the only existing context representation and reasoning mechanism, intended for user-centric web applications, that defines a common framework to integrate domain-specific vocabularies and reasoning rules. Thus, the goal of our ontology is not to provide exhaustive context vocabularies. The motivation of this paper is to illustrate how to extend the SMARTERCONTEXT ontology according to the context awareness requirements of particular domains. For this, we use a *personal web* case study based on the IBM's *smarter commerce* initiative,[1] where the management of context information optimizes the shopper's web experience [5]. For example, by providing retailers with meaningful information about the intents and situations of online shoppers to deliver the proper offer, to the right customer, at the most convenient time.

The remaining sections of this paper are organized as follows. Section 2 introduces the smarter commerce case study and explains, in general terms, the application of our SMARTERCONTEXT solution to this case study. Section 3 presents the semantic web foundations of the SMARTERCONTEXT ontology. Section 4 presents methodological aspects of the definition of the ontology. Section 5 illustrates the application of the SMARTERCONTEXT ontology to context representation in the smarter commerce case study. Section 6 explains the foundational module of the SMARTERCONTEXT ontology. Section 7 presents the modules that extend the ontology for realizing context representation and reasoning in the personal web and the smarter commerce case study. Section 8 explains the context reasoning capabilities supported by the ontology. Section 9 discusses related work. Section 10 posits research challenges and summarizes ongoing work. Finally, Section 11 concludes the paper.

[1] `http://www.ibm.com/smarterplanet/us/en/smarter_commerce/overview`

2 Context Management with SMARTERCONTEXT

2.1 The Smarter Commerce Case Study

Suppose that Norha is a frequent mobile shopping user. To optimize her shopping experience, she registered herself into the SMARTERCONTEXT infrastructure to create her context sphere (the repository of personal context managed by SMARTERCONTEXT). SMARTERCONTEXT gathers relevant context about Norha's situations from different sources such as her mobile devices, and her web interactions. This information is represented using the SMARTERCONTEXT ontology and processed to provide Norha's favorite applications with relevant context about her shopping preferences and situations. Suppose Norha registered the shopping mobile applications of Target,[2] Sears,[3] and Walmart[4] (assuming that these are applications compliant with SMARTERCONTEXT) into her personal context sphere. An application compliant with SMARTERCONTEXT tracks user web interactions and processes Resource Description Framework (RDF) [6] models based on the SMARTERCONTEXT ontology. As a result, the SMARTERCONTEXT infrastructure is now able to gather and provide Norha's relevant context information from and to these retailers. Norha also integrated into her context sphere her shopping list (an application compliant with SMARTERCONTEXT deployed in her mobile device), and her preferred location (e.g., Victoria, BC).

From the very first time Norha browses any of the integrated retailer applications, these applications can take advantage of Norha's personal context to improve her shopping experience. Suppose Norha is browsing the Target's product catalog. Since the application knows Norha's situation and preferences, it suggests relevant products accordingly. Norha can interact with these products through web interactions such as likes, tags, wish lists, and rankings. Product categories involved in these interactions constitute relevant context information that is then integrated into the user's personal context sphere. The SMARTERCONTEXT reasoning engine uses gathered context to infer implicit context facts. In this way SMARTERCONTEXT provides more accurate information about Norha's preferences to authorized applications.

Suppose now Norha is visiting Edmonton and has just arrived at West Edmonton Mall.[5] As soon as she gets into the mall, the smarter commerce application in her mobile device suggests deals available at the stores located in the mall, according to her preferences and shopping list. These stores correspond to those that provide the shopping applications integrated into her context sphere. Moreover, shopping preferences of people in her social network can be taken into account by SMARTERCONTEXT to suggest relevant products available at the mall.

2.2 SMARTERCONTEXT Overview

To manage context information with the goal of improving user shopping experiences as described in the case study, we implemented the SMARTERCONTEXT infrastructure [5]. Our context management solution (i) gathers context from the interactions of

[2] http://www.target.ca
[3] http://www.sears.ca
[4] http://www.walmart.ca/en
[5] http://www.wem.ca

users with web entities (e.g., the interactions of Norha with the products offered in Target's catalog); (ii) integrates this context into the user's personal repository of context information called the *personal context sphere (PCS)*; (iii) reasons why the information is stored in a PCS (e.g., to suggest products related to the products from Sears catalog that the user just added into her wish list); (iv) provides meaningful context to any web application compliant with SMARTERCONTEXT, authorized by the user, and related to the user's current web experience—we call these applications personal web enabled (PWE) applications; and (vi) enables users to delete and modify personal context, as well as to grant and deny context sharing privileges when desired.

The SMARTERCONTEXT infrastructure is composed of a context reasoning engine that relies on the SMARTERCONTEXT ontology and uses semantic web technologies to infer implicit context facts, and a cloud-based service component architecture (SCA) infrastructure that exploits web services to implement context gathering and provisioning. For the smarter commerce case study, SMARTERCONTEXT includes a browser extension for the identification of context providers and consumers such as PWE sites with which the user interacts. These web sites must deploy an interoperability component that enables them to exchange context information with the SMARTERCONTEXT engine. This interoperability component implements two services. The first service is to obtain the context provided by the SMARTERCONTEXT infrastructure. The second is to send context gathered from the interactions of users with web entities to the SMARTERCONTEXT infrastructure. This component includes also two internal methods. The first one keeps track of user interactions (e.g., likes and wishes), and the other one processes RDF/XML context messages.

Figure 1 below provides an overview of our SMARTERCONTEXT solution applied to a smarter commerce case study. The big circle represents the user in her online shopping experience. PWE-Site 1 and 2 are two web sites that are able to communicate with the SMARTERCONTEXT engine. The user authorized SMARTERCONTEXT to gather personal context from her interactions with these web sites, and to provide them with relevant context about her. Suppose user Norha uses a PWE-browser (a browser enabled with the SMARTERCONTEXT extension) to load an online shopping catalog provided by PWE-Site 1 (i.e., Label 1). Suppose the user adds into her wish list a pair of earrings available in this catalog. Since PWE-Site 1 is a context provider authorized by the user, SMARTERCONTEXT gathers meaningful context about Norha's preferences from this interaction (i.e., Label 2 and 3). The SMARTERCONTEXT component deployed at PWE-Site 1 sends this context information in the form of an RDF/XML message to the SMARTERCONTEXT infrastructure, which integrates the gathered context into the user's PCS. The SMARTERCONTEXT engine infers new context facts about Norha's preferences based on the information stored in her context repository. For example, the engine can infer that Norha may be interested in the product category "necklaces" since she added the "earrings" category into her wish list. Suppose now the user interacts with PWE-Site 2 (i.e., Label 4). Since it is an authorized context consumer, SMARTERCONTEXT provides PWE-Site 2 with relevant context about Norha's preferences. This web site can now exploit this information to deliver more relevant shopping offers to the user (i.e., Label 5).

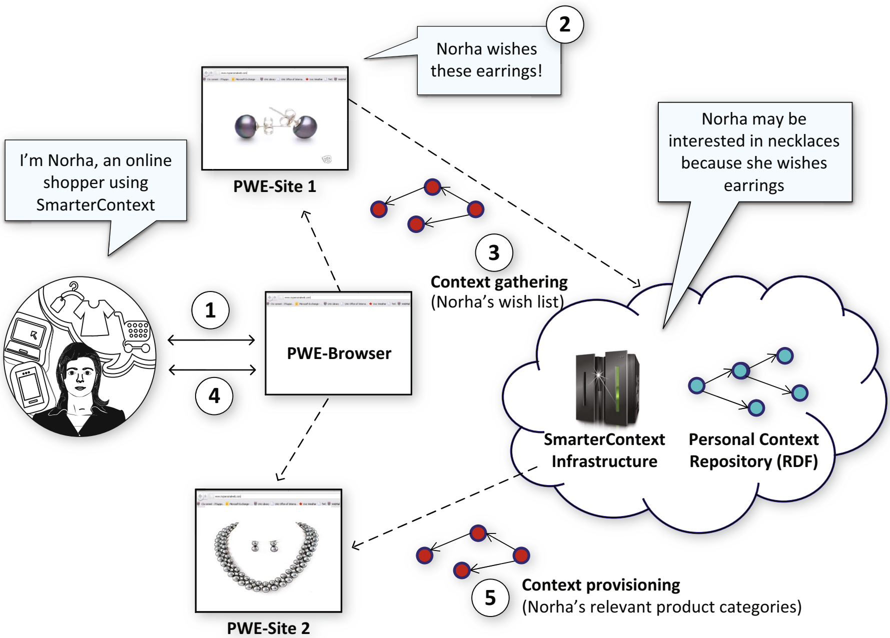

Fig. 1. Our SMARTERCONTEXT solution applied to the smarter commerce case study

Further details on our SMARTERCONTEXT infrastructure and its user-controlled privacy and security mechanisms are available in [5] and [7], respectively.

3 Semantic Web Foundations

The semantic web can be defined as an extension of the web that enables systems to smartly search, combine, and process web data based on the meaning that this data has to humans [8]. This extension exploits the potential of the web since it allows data sharing effectively across the internet [9].

Semantic web technologies provide the means to build models that allow the description of anything in the web, to reason about the knowledge encoded by these models, and to transmit this knowledge among web resources [8]. Our solution exploits semantic web technologies to manage context information as required by the smart internet. First, RDF provides the framework to represent context entities and describe relevant information about them. Thus the SMARTERCONTEXT ontology is based on RDF. Second, RDF, RDF Schema (RDFS) [10], and OWL [11] provide the semantic mechanisms to reason about context entities. Reasoning rules in SMARTERCONTEXT can be defined hierarchically such that more general rules are useful across the corresponding sub-domains. Finally, XML, RDF and OWL provide the data integration and interoperability mechanisms for context gathering and provisioning.

3.1 Linked Data and the Resource Description Framework

The vision of the semantic web relies on linked data, a common framework based on RDF for sharing data and integrating a variety of applications [9,8]. Linked data allows the creation of typed links between data from different sources. This implies data published on the web such that it is machine-processable, it has an explicit meaning, and can be linked to other data sets. SMARTERCONTEXT, supported by linked data, realizes context gathering and provisioning by making context information available to be discovered, and machine-processable since context information is represented in a standardized way. Furthermore, SMARTERCONTEXT realizes context reasoning as its ontology provides explicit semantics that allow inferring implicit context facts.

Linked data uses RDF to describe things in any application domain using typed statements also known as labeled links. Building on top of RDF, the SMARTERCONTEXT ontology provides the entity types (context things), object properties (labeled links between entity types that represent context relationships), and data properties (labeled links between entity attributes and their corresponding values) to describe context entities.

Linked data is based on two fundamental web technologies, Uniform Resource Identifiers (URIs) [12], and the HyperText Transfer Protocol (HTTP) [13]. A URI is a compact sequence of characters that identifies an abstract or physical resource. The HTTP protocol provides a mechanism for retrieving information about entities identified by URIs. RDF is based on the principle that things can be described by making statements about their properties and corresponding values. For this, RDF encodes data in the form of statements defined as *subject*, *predicate*, *object* triples [6]. The subject is the entity the statement is about, the predicate is the property being described about this entity, and the object corresponds to the value of the described property.

Figure 2 depicts a simple RDF statement with corresponding subject, predicate, and object. This statement provides context information about our user's preferred location: "Norha" (the subject) has "preferred location" (the predicate) "Victoria" (the object). Subject, predicate, and object are identified by a URI. For convenience, RDF specifications use a shorthand for referring to URI references (QName). In this way, the full URI is defined by appending the local identifier to the abbreviation (QName prefix). For example, the statement presented in Fig. 2 involves two QName prefixes: `pwc:` to abbreviate the namespace of one of the modules of the SMARTERCONTEXT ontology: `http://smartercontext.org/vocabularies/pwc/v5.0/pwc.owl#`, and `geo:` to abbreviate the namespace of the vocabulary for geographical locations: `http://smartercontext.org/vocabularies/rdf/geo.rdf#`.

Table 1 provides the list of namespaces and corresponding prefixes for the schemas and ontologies used throughout this paper. Protégé [14], the tool used to create and edit ontologies in the SMARTERCONTEXT project, can easily be used to visualize the ontologies described in this paper.

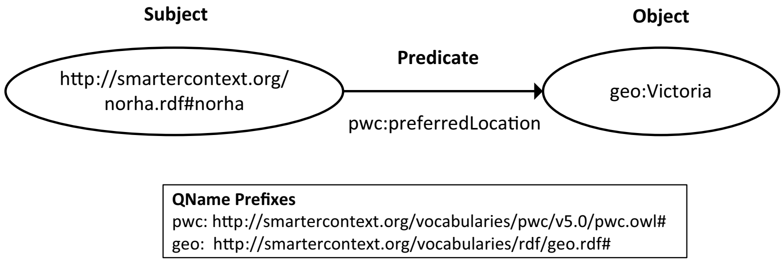

Fig. 2. A simple RDF statement

Table 1. RDF and OWL schemas, and ontologies used throughout the paper

Prefix	Namespace
gc:	smartercontext.org/vocabularies/gc/v5.0/gc.owl#
pwc:	smartercontext.org/vocabularies/pwc/v5.0/pwc.owl#
shopping:	smartercontext.org/vocabularies/shopping/v5.0/shopping.owl#
geo:	smartercontext.org/vocabularies/rdf/geo.rdf#
google:	smartercontext.org/vocabularies/rdf/googleproducts.rdf#
deals:	smartercontext.org/vocabularies/rdf/dealcategories.owl#
rdf:	www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs:	www.w3.org/2000/01/rdf-schema#
owl:	www.w3.org/2002/07/owl#

3.2 Vocabularies

Semantic web vocabularies are collections of classes and properties expressed in RDF using types from the RDF Vocabulary Definition Language (RDFS) [10] and the Web Ontology Language (OWL) [11].

RDF Vocabulary Definition Language (RDFS). RDF Schema is a semantic extension of RDF that defines classes and properties used to describe classes, properties, and other RDF resources. Tables 2 and 3 summarize the classes and properties from the RDFS specification [10] used in the SMARTERCONTEXT ontology.

The Web Ontology Language (OWL). RDFS is suitable for modeling simple ontologies and has limited knowledge inference capabilities [8]. To model more complex knowledge, the semantic web provides OWL, an expressive representation language based on formal logic. OWL is used to model ontologies. An OWL ontology is a set of classes, properties, and individuals useful to describe entities and the relationships among them in a particular application domain. Classes are instances of `owl:Class`, which is a subclass of `rdfs:Class`. Therefore, as described in Table 2, OWL classes

Table 2. RDF Schema classes used in our SMARTERCONTEXT ontology

Class	Description
rdfs:Resource	Any entity described by RDF — e.g., user Norha.
rdfs:Class	The class of resources that are RDF classes — e.g., a class User that defines the entity Norha.
rdfs:Literal	The class of resources that are values such as strings or integers. Literals may be typed or untyped — e.g., the values for the age and the address of user Norha.
rdfs:Datatype	Any datatype defined in the XML Schema [15].

Table 3. RDF Schema [10] properties used in our SMARTERCONTEXT ontology

Property	Description
rdfs:range	Defines the universe of possible values of a property — e.g., the possible values of the property `pwc:preferredLocation` correspond to entities of type *Location* context.
rdfs:domain	States that any resource with a given property is an instance of one or more classes — e.g., any resource that has a preferred location is an instance of class *User*.
rdf:type	States that a resource is an instance of a class — e.g. Norha is an instance of type *User*.
rdfs:subClassOf	States that all the instances of a class are instances of another one — e.g., every instance of class User is an instance of class *HumanEntity*.
rdfs:sub PropertyOf	States that all the resources related by a property are also related by another one — e.g., if *Norha* is related to *Peter* by the property *daughterOf*, and *daughter of* is a subproperty of *relative of*, *Norha* is related to *Peter* by the property *relative of*.

are RDF resources of type class. Individuals correspond to instances of classes. OWL defines two types of properties, abstract properties and concrete properties. Abstract properties relate individuals with individuals, whereas concrete properties link individuals with data values. Both are subtypes of *rdf:Property*.

The SMARTERCONTEXT ontology is based on RDF and a subset of the OWL-Lite [16] specification. Taking into account that pure RDFS is not sufficient for context reasoning as envisioned in SMARTERCONTEXT, we decided to use the simpler version of OWL called OWL-lite which provides enough support for context representation and reasoning. Table 4 describes the OWL-Lite properties used in our SMARTERCONTEXT ontology.

Table 4. OWL-Lite [16] properties used in our SMARTERCONTEXT ontology

Feature	Description
owl:inverseOf	If properties P_1 and P_2 are inverse, then if X is related to Y by P_2, then Y is related to X by P_1 — e.g., properties *hosts* and *hosted by* are inverse. Thus, if CASCON 2012 is *hosted by* Hilton Markham, then Hilton Markham *hosts* CASCON 2012.
owl:Transitive Property	If a property P is transitive, then if X is related to Y by P, and Y is related to Z by P, then X is related to Z by P — e.g., *located in* is a transitive property. If Norha is *located in* Victoria, and Victoria is *located in* British Columbia, then Norha is *located in* British Columbia.
owl:Functional Property	A property that has at most one value — e.g., the year a human entity was born.
owl:Symmetric Property	If a property P is symmetric and X is related to Y by P, then Y is related to X by P — e.g., the property *near to* is symmetric. If Victoria is *near to* Vancouver, then Vancouver is *near to* Victoria.

4 Introduction to the SMARTERCONTEXT Ontology

The SMARTERCONTEXT ontology is an RDF-based vocabulary defined to represent explicit context information, and to reason about these context representations to derive implicit context facts at runtime. The version of the SMARTERCONTEXT ontology presented in this paper relies on OWL-Lite to reason about context information using formal logic [16]. That is, context reasoning capabilities are based on the RDFS classes presented in Table 2, the RDFS properties presented in Table 3, and the subset of OWL-Lite properties presented in Table 4.

4.1 Methodological Aspects

The genesis of our context manager and the SMARTERCONTEXT ontology is an extensive survey on context modeling and context management approaches in different problem domains of context-aware computing [2]. The motivation of this systematic literature review, from the perspective of the smart internet, was the identification of context modeling and context management requirements to support context-awareness as required by smart interactions and services. As a result, we proposed a general classification of context information. This general classification, known as the *General Context (GC)* taxonomy, constitutes the fundamental module of the SMARTERCONTEXT ontology.

4.2 Requirements Analysis

We define the requirements for context representation in user-centric smart internet applications as follows:

RQ-i. Context information must be gathered and provided in an interoperable way.

RQ-ii. It must be possible to represent context entities, the relationships among them, the properties that characterize their situation, and the relationships between these entities and the user.

RQ-iii. Timeliness modeling must be supported (i.e., the representation of past, present and future situations).

RQ-iv. Context representations must be able to adapt at runtime according to changes in the situation of users and systems. That is, context entities may appear, disappear or be modified dynamically without affecting the relevance of the context management infrastructure.

Regarding RQ-i, the knowledge represented in a semantic format is better suited for interoperation from the perspective of systems and knowledge sources [8]. Concerning requirement RQ-ii, RDFS and OWL-Lite provide sufficient expressiveness to characterize context types with corresponding properties, and to represent context relationships, constraints and granularity levels. Concerning RQ-iii and RQ-iv, context models based on RDF graphs support the representation of context data over time, and can easily be modified at runtime to add or delete context facts according to changes in context situations [5,17].

4.3 Extensibility and Modularity

Modularization, as in many other domains, is a best practice in ontology design [8]. The increasing size and complexity of context models require collaborative design. Moreover, the design of loosely coupled ontologies facilitates their processing, maintenance and evolution. Modular ontologies also guarantee privacy and security requirements since it is easier to control the level of exposure of sensible data [7].

We designed SMARTERCONTEXT as a modular and extensible ontology. Its foundational module, *general context (GC)*, enables context representation and reasoning for any problem domain of the smart internet. Because of its modular structure, the SMARTERCONTEXT ontology supports vertical and horizontal extensibility. *Vertical extensibility* makes the SMARTERCONTEXT ontology applicable to different problem domains. It is realized by defining more specialized modules that inherit from the GC module or other modules derived from GC. The application of the SMARTERCONTEXT ontology to a particular domain may imply the definition of several hierarchical levels. For example, to support context-awareness in the personal web (PW) we derived from GC the *personal web context (PWC)* module. The PWC module supports context representation and reasoning in any problem domain of the PW. To apply SMARTERCONTEXT to a particular application of the PW, the recommended practice is to extend the PWC module further by defining more particular context types and context reasoning rules according to the specific domain. For example, we derived from PWC the *shopping* module to support context representation and reasoning in our smarter commerce case study [5]. *Horizontal extensibility* is realized by importing existing vocabularies into any of the ontology's modules.

Table 5. RDF triples that illustrate the personal context sphere for user Norha

#	Subject	Predicate	Object
1	google:XBox_360_Consoles	rdf:type	shopping:ProductService Category
2	gabriel.rdf#gabriel	rdf:type	gc:HumanEntity
3	google:Earrings	rdf:type	shopping:ProductService Category
4	geo:Victoria	rdf:type	gc:GeoLocation
5	deals:Gyms_&_ Fitness_Centers	rdf:type	shopping:ProductService Category
6	google:Electric_Grills	rdf:type	shopping:ProductService Category
7	google:Tennis_Shoes	rdf:type	shopping:ProductService Category
8	http://www.wem.ca	gc:geoLocation Classification	"Place"^^xsd:string
9	http://www.wem.ca	rdf:type	gc:PhysicalLocation
10	http://www.wem.ca	rdf:type	gc:PhysicalLocation
11	norha.rdf#norha	pwc:isInterestedIn	deals:Gyms_&_ Fitness_Centers
12	norha.rdf#norha	pwc:hasIntegrated	http://www.sears.ca
13	norha.rdf#norha	shopping:toBuy	google:Electric_Grills
14	norha.rdf#norha	gc:locatedIn	http://www.wem.ca
15	norha.rdf#norha	pwc:marriedTo	gabriel.rdf#gabriel
16	norha.rdf#norha	rdf:type	pwc:User
17	norha.rdf#norha	pwc:likes	google:XBox_360_Consoles
18	norha.rdf#norha	pwc:preferredLocation	geo:Victoria
19	norha.rdf#norha	shopping:wishes	google:Earrings
20	norha.rdf#norha	pwc:hasIntegrated	http://www.target.ca
21	norha.rdf#norha	pwc:hasIntegrated	http://www.walmart.ca.en
22	norha.rdf#norha	shopping:toBuy	google:Tennis_Shoes
23	norha.rdf#norha	pwc:colleagueOf	tatiana.rdf#tatiana
24	http://www.walmart.ca/en	rdf:type	pwc:PWESite
25	http://www.sears.ca	rdf:type	pwc:PWESite
26	http://www.target.ca	rdf:type	pwc:PWESite

5 Context Representation in the Personal Web

Context information in our context management solution is represented in the form of RDF graphs where resources and predicates (nodes and arcs) are compliant with the types defined in the SMARTERCONTEXT ontology. In the case study presented in this paper these types correspond to the classes, object properties, and data properties defined in the ontology's modules: GC, PWC, and Shopping, including their horizontal extensions.

A partial version of a context sphere for user Norha in the scenario described in Sect 2.1 is available in `http://smartercontext.org/examples/norha.rdf`. This context model is composed of 26 SMARTERCONTEXT triples detailed in Table 5. As explained in Section 3.1, each triple represents an RDF statement, context facts in SMARTERCONTEXT, defined by a subject, a predicate, and an object.

The World Wide Web Consortium[6] (W3C) provides an RDF validation service[7] useful to visualize small RDF graphs. It is possible to visualize the exemplar of user Norha's context repository used in this paper by copying the RDF/XML contents to the "Check by Direct Input" field of the validator, or by entering the URL of the model (`http://smartercontext.org/examples/norha.rdf`) in the field "Check by URI" of the validator.

6 The General Context (GC) Module

The GC module defines context types (classes), abstract properties (relationships between classes), and concrete properties (links between attributes of individuals and their corresponding values) applicable to any problem domain, for instance the smart internet.

6.1 Context Entities in the GC Module

Table 6 details the entities defined in the GC module. For each class, it presents the corresponding context entity type (column *Entity*), the class the entity is derived from (column *Superclass*), and a description of the role that the entity plays in the SMARTERCONTEXT ontology (column *Description*). This column schema is used to describe the class types of the ontologies presented in the subsequent sections of this paper.

6.2 Object Properties in the GC Module

Table 7 presents the object properties (abstract properties) defined in the GC module. GC's object properties constitute context relationships between context entities defined in the GC module or in any of its extensions. For each object property, this table presents in column *Domain* the values of the domain, in column *Range* the values of the range, in column *Features* whether the property is transitive (T), functional (F), or symmetric

[6] `http://www.w3.org/`

[7] `http://www.w3.org/RDF/Validator`

Table 6. Context entities defined in the GC module of the SMARTERCONTEXT ontology

Entity (Class)	Superclass	Description
ContextEntity	owl:Thing	The superclass of any context type.
ActivityContext	ContextEntity	Actions and tasks performed by an object - e.g., attending a meeting.
IndividualContext	ContextEntity	Anything that can be observed about an isolated object - e.g., the object location.
ArtificialEntity	IndividualContext	Entities resulting from human actions or technical processes - e.g., buildings, hardware and software configurations.
GroupEntity	IndividualContext	Groups of subjects that share common characteristics but not necessarily interact with each other - e.g., Canadian women.
HumanEntity	IndividualContext	Any information about a person's behavior, preferences, characteristics and way of interacting with a system - e.g., an online shopper.
NaturalEntity	IndividualContext	Living and non-living entities which are not the direct result of any human activity - e.g., weather conditions.
LocationContext	ContextEntity	The place of settlement or activity of an object.
PhysicalLocation	LocationContext	A physical place of settlement or activity of an object - e.g., University of Victoria.
GeoLocation	PhysicalLocation	The latitude and altitude that describe a physical location.
VirtualLocation	LocationContext	Location describable by a URI - e.g., namespace of the SMARTERCONTEXT ontology.
Endpoint	VirtualLocation	A URI that identifies the location of a computational resource - e.g., the SOAP address of a context sensor exposed as a service.
TimeContext	ContextEntity	Provides context about a specific date and time, but also categorical information such as holidays, working days, and meeting schedules - e.g., Boxing Day.
DefiniteTime	TimeContext	Represents time frames with specific begin and end points (i.e., the duration of a conference).
IndefiniteTime	TimeContext	Expresses a recurrent event which is happening while another situation is taking place. It is not possible to know its duration in advance -e.g., the time a service is online.

(S), and in column *Inverse Of* the properties that are inverse of the described property. This column schema is used to describe object properties in the PWC and Shopping modules of the SMARTERCONTEXT ontology.

The `associationRelationship` object property represents aggregation and association context relationships (different than functional and social relations). Both its domain and range correspond to entities of the type `contextEntity`. `locationRelationship` includes any object property with range equal to the `LocationContext` type. GC defines no domain for the `LocationContext` property as it may depend on the specific application domain. `hostedBy` and `locatedIn` are sub-properties of `locationRelationship`. The value of `hostedBy` in a triple represents a `LocationContext` entity that hosts the `ActivityContext` entity represented by the subject. The value of `locatedIn` represents the location where the subject (an `IndividualContext` or `LocationContext` entity) is located in. The `functionalRelationship` object property refers to information about the usage that an object can make of another. As indicated by its domain and range, functional relationships can exist between any pair of context entities. The value of the `hosts` property, which inherits from `functionalRelationship`, corresponds to a scheduled event that has place in the `LocationContext` entity represented by the subject. Finally, the `socialRelationship` object property emerges from the interrelation between individuals of type `HumanEntity` and `GroupEntity`. Samples of this relational context are affiliations, colleagues, and customers.

Table 7. Object properties defined in the GC module of the SMARTERCONTEXT ontology. The *T* in column *Features* stands for owl:TransitiveProperty.

Property	Domain	Range	Features	Inverse Of
association Relationship	ContextEntity	ContextEntity	-	-
location Relationship	-	LocationContext	-	-
hostedBy	ActivityContext	LocationContext	-	hosts
locatedIn	IndividualContext LocationContext	LocationContext	T	-
functional Relationship	ContextEntity	ContextEntity	-	-
hosts	LocationContext	ActivityContext	-	hostedBy
social Relationship	GroupEntity HumanEntity	GroupEntity HumanEntity	-	-

6.3 Data Properties in the GC Module

Table 8 details the data properties (concrete properties) defined in the GC module. Data properties allow the description of context attributes (i.e., characteristics of context entities). All of these properties correspond to functional properties, that is, properties that have at most one value. The domain corresponds to the context entity type for which the data property is defined. The range details the valid data types for the values of the properties. In many cases, the range of a data property is restricted to a set of specific values. Such is the case of the `geoLocationClassification` data property.

6.4 Horizontal Extension in the GC Module

Foundational elements defined in the GC module can be extended horizontally by importing concrete vocabularies. In this case study we extended the GC module by defining a vocabulary to characterize `GeoLocation` entities and the relationships among them. Any other semantic web geographical vocabulary can be used for this extension.

Table 8. Data properties defined in the GC module of the SMARTERCONTEXT ontology. All of the properties in this table correspond to functional properties.

Property	Domain	Range	Value Description
address	GeoLocation	xsd:string	Corresponds to a String literal that denotes the exact location of a GeoLocation context entity.
endDateTime	DefiniteTime	xsd:dateTime	A dateTime value that denotes the end time of DefiniteTime context entity (the last value of the time interval).
geoLocation Classification	GeoLocation	“City”, “Country”, “Neighborhood”, “Place”, “Region”	Classifies GeoLocation context types.
latitude	GeoLocation	xsd:string	The angular distance north or south of the Equator, in degrees, minutes, and seconds of a GeoLocation context entity.
longitude	GeoLocation	xsd:string	The angular distance, in degrees, minutes, and seconds, of GeoLocation context entity east or west of the Prime (Greenwich) Meridian.
startDateTime	DefiniteTime	xsd:dateTime	A dateTime value that denotes the beginning of a DefiniteTime context entity (the initial value of the time interval).
zipCode	GeoLocation	xsd:string	A string value that corresponds to the postal code of the GeoLocation entity represented by the subject.

7 SmarterContext in the Personal Web

7.1 The Personal Web Context (PWC) Module

The PWC module extends the GC module vertically to define context types, object properties and data properties required to represent and reason about context information in context-aware applications within the personal web domain.

Context Entities in the PWC Module. Table 9 details the entities defined in the PWC module. `PWConcern` allows smart interactions and services to understand the nature of users mocs at a specific time (e.g., whether the user is surfing the web for shopping or social activities). The `PWConcern` entity defines seven categories of personal web concerns: *Academic, Business, Entertainment, Healthcare, Shopping, Social,* and *Travel.*

Table 9. Context entities defined in the PWC module of the SmarterContext ontology

Entity (Class)	Superclass	Description
PWConcern	gc:ActivityContext	Classifies web resources and activities a user performs in the web - e.g., shopping, academic, healthcare.
ScheduledEvent	gc:ActivityContext	A calendar event defined in a personal agenda - e.g., a business trip.
PhysicalEntity	gc:ArtificialEntity	A context entity that is not available as a web entity. E.g., the user's preferred currency.
WebResource	gc:ArtificialEntity	Web elements the user interacts with such as web sites, and web services - e.g., Walmart's shopping site.
PWESite	WebResource	Represents a web site compliant with the SmarterContext framework [5].
WebEntity	WebResource	Any entity available on the web different than PWE sites and web services - E.g., products or services offered online, a personal health record.
WebService	WebResource	Any web service relevant to the user - e.g., a service for payments with credit cards.
User	gc:HumanEntity	Refers to any information about the user's behavior and preferences -e.g., security profiles, language preferences, and personal information.

Object Properties in the PWC Module. Table 10 details the object properties (abstract properties) defined in the PWC module. Column *Features* indicates whether the property is transitive (T), functional (F), or symmetric (S). PWC object properties, which extend from the object properties defined in the GC module, allow the definition of context relationships between context entities defined in the PWC module or in any of its extensions.

The concerns property associates context entities of type ScheduledEvent, gc:NaturalEntity, gc:ArtificialEntity, gc:LocationContext, and gc:GroupEntity with relevant categories defined as entities of type PWConcern (e.g., a personal calendar event associated with a shopping concern). hasIntegrated allows the integration of any instance of type IndividualContext to the user's context sphere (e.g., a personal agenda application integrated through a web service). isNearTo associates two entities of type gc:GeoLocation as close to each other (within a short distance). Since property isNearTo is symmetric, it applies to both entities. Property preferredLocation defines a gc:GeoLocation entity as the user's favorite place of settlement. As it is a functional property, each user can have one preferred location at most. concerns, and hasIntegrated are sub-properties of gc:associationRelationship. isNearTo, and preferredLocation are sub-properties of gc:locationRelationship, which is sub-property of gc:associationRelationship.

Property identifiedBy is useful for identifying context entities of type WebResource. The value of this property is an entity of type gc:Endpoint. For example a shopping web site identified by its URL http://www.amazon.ca/. scheduledFor is used to define the schedule of calendar events. Both identifiedBy and scheduledFor are functional properties and inherit from gc:functionalRelationship. Another PWC object property that extends from gc:functionalRelationship is userInteraction. This property is absolutely crucial for context integration in the personal web since the user is the one who knows about web entities and their relationship with her own situation. User interactions provide the means to identify context entities relevant to the user's situation throughout her web experience. For example, through a simple interaction such as *liking* a product, the user provides the SMARTERCONTEXT infrastructure with relevant information about her preferences. The current version of the SMARTERCONTEXT ontology subdivides user interactions defined in the PWC module in the following object properties: dislikes, isInterestedIn, likes, ranked, and tagged.

Social context relationships (gc:socialRelationship) emerge from the interrelation between entities of type GroupEntity and HumanEntity. The PWC module defines the following object properties to represent social relationships in any application domain of the personal web: affiliatedWith, associates, colleagueOf, engagedTo, friendOf, and relativeOf, which is subdivided in childOf, marriedTo, parentOf, and siblingOf. affiliatedWith and associates are inverse. colleagueOf, engagedTo, and friendOf are symmetric properties and apply between two entities of type gc:HumanEntity. childOf and parentOf are inverse properties.

Data Properties in the PWC Module. Table 11 details the data properties (concrete properties) defined in the PWC module. The data

Table 10. Object properties defined in the PWC module of the SMARTERCONTEXT ontology. The *S* and the *F* in column *Features* stand for owl:SymmetricProperty and owl:FunctionalProperty, respectively.

Property	Domain	Range	Features	Inverse Of
concerns	ScheduledEvent gc:NaturalEntity gc:ArtificialEntity gc:LocationContext gc:GroupEntity	PWConcern	-	-
hasIntegrated	User	gc:IndividualContext	-	-
isNearTo	gc:GeoLocation	gc:GeoLocation	S	-
preferredLocation	User	gc:GeoLocation	F	-
identifiedBy	WebResource	Endpoint	F	-
scheduledFor	ScheduledEvent	gc:DefiniteTime	F	-
userInteraction	gc:HumanEntity	gc:IndividualContext	-	-
dislikes	User	gc:IndividualContext	-	-
isInterestedIn	User	gc:IndividualContext	-	-
likes	User	gc:IndividualContext	-	-
ranked	User	gc:IndividualContext	-	-
tagged	User	gc:IndividualContext	-	-
affiliatedWith	gc:HumanEntity	gc:GroupEntity	-	associates
associates	gc:GroupEntity	gc:HumanEntity	-	affiliatedWith
colleagueOf	gc:HumanEntity	gc:HumanEntity	S	-
engagedTo	gc:HumanEntity	gc:HumanEntity	S	-
friendOf	gc:HumanEntity	gc:HumanEntity	S	-
relativeOf	gc:HumanEntity	gc:HumanEntity	S	-
childOf	gc:HumanEntity	gc:HumanEntity	-	parentOf
marriedTo	gc:HumanEntity	gc:HumanEntity	S	-
parentOf	gc:HumanEntity	gc:HumanEntity	-	childOf
siblingOf	gc:HumanEntity	gc:HumanEntity	S	-

properties `birthYear`, `emailAccount`, `givenName`, `hasGender`, `lastName`, and `preferredLanguage` define attributes of context entities of type `gc:HumanEntity`. `rankingValue` rates any entity of type `gc:IndividualContext`. It is used together with the `ranked` object property. `scheduledEventDescription` and `scheduledEventTittle` describe attributes of calendar events (i.e., instances of type `ScheduledEvent`).

Table 11. Data properties defined in the PWC module of the SmarterContext ontology

Property	Domain	Range	Value Description
birthYear	gc:HumanEntity	xsd:int	Functional. The year a human entity was born.
emailAccount	gc:HumanEntity	xsd:string	An email account associated with the human entity represented by the subject.
givenName	gc:HumanEntity	xsd:string	Functional. The given name of the human entity represented by the subject.
hasGender	gc:HumanEntity	"Female", "Male", "NotSpecified"	Functional. The gender of the human entity represented by the subject.
lastName	gc:HumanEntity	xsd:string	Functional. The last name of the human entity represented by the subject.
preferred Language	gc:HumanEntity	xsd:string	The preferred language of the human entity represented by the subject.
rankingValue	gc:Individual Context	xsd:int	Functional. The ranking value assigned by the user to the entity represented by the subject.
scheduledEvent Description	ScheduledEvent	xsd:string	Functional. The description of the scheduled event represented by the subject.
scheduledEvent Title	ScheduledEvent	xsd:string	Functional. The title of the scheduled event represented by the subject.

7.2 The Shopping Module

The SmarterContext Shopping module is an extension of the PWC module that supports context representation and reasoning in smarter commerce applications based on the PW. This section presents how we extended the PWC module, horizontally and vertically, to realize user-centric shopping interactions in our smarter commerce case study [5].

Context Entities in the Shopping Module. Table 12 details the classes defined as context entity types in the shopping module.

Object Properties in the Shopping Module. Table 13 details the types required to represent context relationships in the Shopping module. The `relatedProductOrService` object property, which extends from `gc:associationRelationship`, denotes that two product or service categories are related to each other (e.g., complementary products such as necklaces and earrings). `preferredCurrency`, `preferredDeliveryMethod`, and

Table 12. Context entities defined in the Shopping module of the SMARTERCONTEXT ontology

Entity (Class)	Superclass	Description
Currency	pwc:PhysicalEntity	Represents one of the user's preferred currencies - e.g., CAD, USD.
Delivery Method	pwc:PhysicalEntity	Represents one of the user's preferred delivery methods - e.g., Fedex.
Payment Method	pwc:PhysicalEntity	Represents one of the user's preferred payment methods - e.g., credit card, PayPal.
ProductService Category	pwc:WebEntity	Denotes a product or a service category offered online - e.g., Clothing, Electronics.

`preferredPaymentMethod` extend from `gc:functionalRelationship` and associate currencies, delivery methods and payment methods to the user. Four new types of user interactions extend `pwc:userInteraction` in the Shopping module. The first one, `doesNotWish`, indicates that the product or service category represented by the object cannot be part of the user's wish list. The second one, `purchased`, allows the SMARTERCONTEXT infrastructure to identify products the user purchased during her interactions with a particular shopping site. The third one, `toBuy`, indicates that the product or service category represented by the object is in the user's shopping list. Finally, the `wishes` object property represents that the corresponding product or service category was added by the user into her wish list.

Table 13. Object properties defined in the Shopping module of the SMARTERCONTEXT ontology. The *S* in column *Features* stands for owl:SymmetricProperty.

Property	Domain	Range	Features	Inverse Of
relatedProduct orService	ProductService Category	ProductService Category	S	-
preferred Currency	pwc:User	Currency	-	-
preferred DeliveryMethod	pwc:User	DeliveryMethod	-	-
preferred PaymentMethod	pwc:User	PaymentMethod	-	-
doesNotWish	pwc:User	ProductService Category	-	-
purchased	pwc:User	ProductService Category	-	-
toBuy	pwc:User	ProductService Category	-	-
wishes	pwc:User	ProductService Category	-	-

Table 14. Data properties defined in the Shopping module of the SMARTERCONTEXT ontology

Property	Domain	Range	Value Description
billingAddress	PaymentMethod	xsd:string	Functional. A string that represents the billing address of a PaymentMethod context entity.
cardNumber	PaymentMethod	xsd:string	Functional. A string that represents the card number of a PaymentMethod context entity.
expiration Month	PaymentMethod	xsd:int	Functional. An int that represents the expiration month of a PaymentMethod context entity.
expirationYear	PaymentMethod	xsd:int	Functional. An int that represents the billing address of a PaymentMethod context entity.
nameOnCard	PaymentMethod	xsd:string	Functional. A string that represents the name on card of a PaymentMethod context entity.
payment MethodType	PaymentMethod	xsd:string	Functional. A string that represents the type of a payment method - e.g., Visa, Mastercard, PayPal.
targetedFor Gender	Product ServiceCategory	Female, Male, None	Functional. A string that indicates whether the product or service category is intended for a particular gender.
verification Number	PaymentMethod	xsd:string	Functional. A string that represents the security number of a PaymentMethod context entity.

Data Properties in the Shopping Module. Table 14 details the data properties (concrete properties) that allow the definition of context attributes for context entities in the Shopping module.

Horizontal Extension in the Shopping Module. The Shopping module of the SMARTERCONTEXT ontology is extended horizontally by importing two RDF vocabularies that characterize products and services. Both vocabularies extend from the `ProductServiceCategory` context type. The first vocabulary corresponds to the *Google Product Taxonomy* [18]. This taxonomy categorizes products in Google's search results. Google provides this taxonomy in two formats, as a plain text file and as a spreadsheet. We converted this hierarchical set of product categories into an RDF vocabulary. The second vocabulary corresponds to the *Groupon Deal Categories* [19]. This taxonomy, available in JSON, XML and spreadsheet formats, provides the complete set of categories used by Groupon to characterize deals offered to users via email. To integrate this taxonomy into the Shopping module of the SMARTERCONTEXT ontology, we generated it as an RDF vocabulary [20].

8 Context Reasoning with the SMARTERCONTEXT Ontology

Context reasoning in the SMARTERCONTEXT framework relies on *deduction rules* supported by the RDFS specification and a subset of the axioms defined in OWL-Lite. Besides standard RDFS and OWL-Lite rules, SMARTERCONTEXT allows the definition of particular reasoning rules according to the problem domain. The definition of domain-dependent reasoning rules is part of the vertical extension capabilities of SMARTERCONTEXT.

The following two sub-sections present selected rules used in our smarter commerce case study to infer context facts about the preferences and situations of user Norha in the shopping scenario described in Section 2.1. Using RDF graph representations of context facts, we illustrate how each rule is applied by the SMARTERCONTEXT engine to infer implicit context facts which are represented by dashed arcs. Explicit context facts about user Norha are borrowed from the partial view of her context sphere[8] detailed in Table 5.

8.1 RDFS and OWL-Lite Deduction Rules

Jena[9] is the semantic web platform that supports context reasoning in SMARTERCONTEXT. Context reasoning rules in Jena are defined as a set of premises, a list of conclusions, and an optional name and optional direction. Each term of a Jena rule corresponds to either a triple pattern, an extended triple pattern, or a call to a built-in function [21]. This sub-section illustrates the application of standard RDFS and OWL-Lite axioms to context reasoning with the SMARTERCONTEXT ontology.

Reasoning from Subclasses. The following deduction rules exploit the semantic characteristics of the `rdfs:subClassOf` object property.

Rule 1 $(?A\ rdfs:subClassOf\ ?B), (?v\ rdf:type\ ?A) \rightarrow (?v\ rdf:type\ ?B)$

Example:

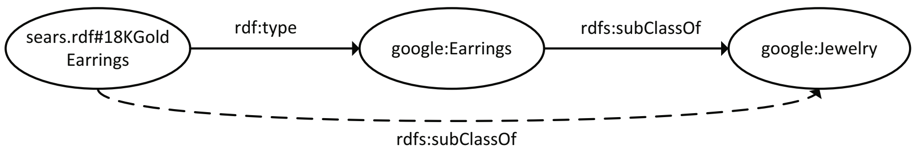

Fig. 3. Inferring implicit context facts with Rule 1

Rule 1 enables the inheritance of a resource's membership in a class A to the superclasses of A. In the example, since the product category `google:Earrings` is a subclass of the product category `google:Jewelry`, and the concrete product `sears.rdf#18KGoldEarrings` is an instance of `google:Earrings`, then this product is also an instance of `google:Jewelry`. An application of this rule to

[8] `http://smartercontext.org/examples/norha.rdf`
[9] `http://jena.sourceforge.net/inference`

smarter commerce is the inference of product and service preferences from the interactions of a user with particular products. For example, knowing that the user has earrings in her wish list (cf. triple 19 in Table 5), it is possible to infer that she would be interested in other jewelry categories.

Rule 2 $(?A\ rdfs : subClassOf\ ?B), (?B\ rdfs : subClassOf\ ?C) \rightarrow (?A\ rdfs : subClassOf\ ?C)$

Example:

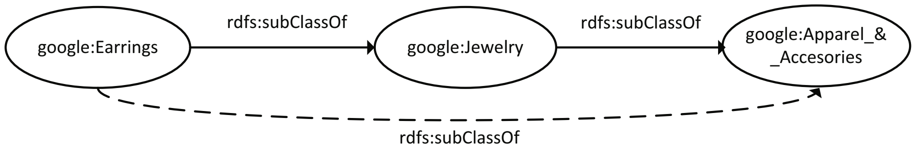

Fig. 4. Inferring implicit context facts with Rule 2

Rule 2 implements the transitivity of the `rdfs:subClassOf` object property. The example of this rule states that since `google:Earrings` is a subclass of `google:Jewelry`, and `google:Jewelry` is a subclass of `google:Apparel_&_Accessories`, then `google:Earrings` is a subclass also of `google:Apparel_&_Accessories`. In our shopping scenario, rules 1 and 2 can be combined to infer that the user may be interested in products of the category apparel & accessories, given that earrings is in her wish list.

Reasoning from Subproperties. The following deduction rules exploit the semantic characteristics of the `rdfs:subPropertyOf` object property.

Rule 3 $(?A\ rdfs : subPropertyOf\ ?B), (?v\ ?A\ ?y) \rightarrow (?v\ ?B\ ?y)$

Example:

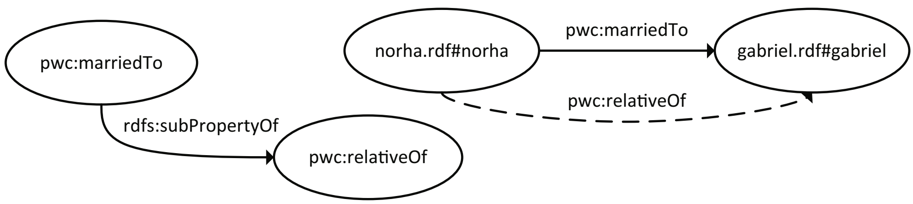

Fig. 5. Inferring implicit context facts with Rule 3

Rule 3 states that any triple with a predicate defined by a property A is also valid for the predicates defined by the superproperties of A. The example illustrates this rule using properties that correspond to family relationships between human entities. Shopping preferences are undeniable affected by the preferences and needs of the shopper's close family. The SMARTERCONTEXT ontology defines `pwc:marriedTo` as a subproperty of `pwc:relativeOf`. In the example, since Norha is married to Gabriel, SMARTERCONTEXT will infer that Norha is a relative of Gabriel.

Rule 4 $(?A\ rdfs : subPropertyOf\ ?B), (?B\ rdfs : subPropertyOf\ ?C) \rightarrow (?A\ rdfs : subPropertyOf\ ?C)$

Example:

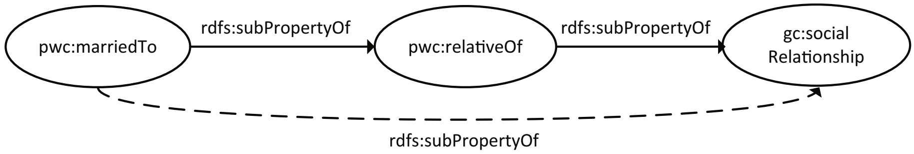

Fig. 6. Inferring implicit context facts with Rule 4

Rule 4 implements the transitivity of the `rdfs:subPropertyOf` object property. The example of this rule states that since `pwc:marriedTo` is a subproperty of `pwc:relativeOf`, and `pwc:relativeOf` is a subproperty of `gc:socialRelationship`, it is possible to infer that any pair of human entities that are married to each other, are not only relatives of each other, but also are socially related to each other. By the combination of rules 3 and 4 for Norha's context sphere, it is possible to infer that a social relationship holds between her and the human entity `gabriel.rdf#gabriel`.

Reasoning from Property Restrictions. The following deduction rules exploit the semantic characteristics of `rdfs:domain` and `rdfs:range`.

Rule 5 $(?A\ rdfs : domain\ ?B), (?u\ ?A\ ?y) \rightarrow (?u\ rdf : type\ ?A)$

Example:

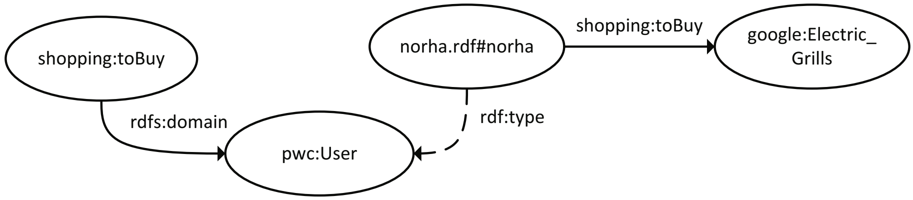

Fig. 7. Inferring implicit context facts with Rule 5

As explained in Section3.2, the domain of a property in SMARTERCONTEXT defines the valid context types for the subjects of the triples where this property acts as the predicate. Therefore, Rule 5 is useful to infer from a triple, the type of the subject context entity by looking at the domain of the predicate. For example, the SMARTERCONTEXT ontology defines the context type `pwc:User` as the domain of the property `shopping:toBuy`. Therefore, from the context fact `(norha.rdf#norha shopping:toBuy google:Electric_Grills)`, SMARTERCONTEXT infers that `norha.rdf#norha` is an entity of type `pwc:User`.

Rule 6 $(?A\ rdfs:range\ ?B), (?u\ ?A\ ?y) \rightarrow (?y\ rdf:type\ ?B)$

Example:

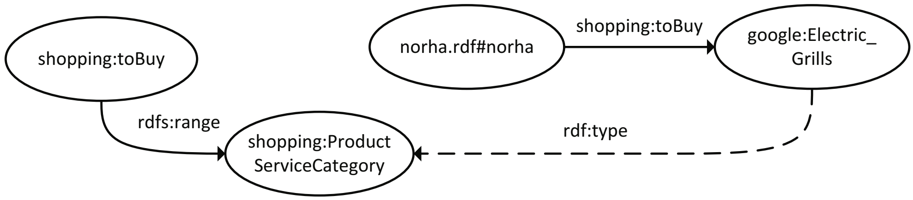

Fig. 8. Inferring implicit context facts with Rule 6

Rule 6 allows the inference of class memberships for objects of triples. The range of a property in a particular triple defines the valid types for the objects of the triple. In the example of this rule it is possible to infer that `google:Electric_Grills` is an entity of type `shopping:ProductServiceCategory`, given that the latter is the range of the `shopping:toBuy` property in SMARTERCONTEXT. Rules 5 and 6 are useful in smarter commerce scenarios for instance to recommend product or service categories by inferring the types of particular products the user has interacted with, and applying complementary rules such as rules 1 and 2.

Reasoning from Transitive Properties. The following deduction rules exploit the semantic characteristics of transitive properties in OWL-Lite.

Rule 7 $(?A\ rdf:Type\ owl:TransitiveProperty), (?u\ ?A\ ?v), (?v\ ?A\ ?x) \rightarrow (?u\ ?A\ ?x)$

Example:

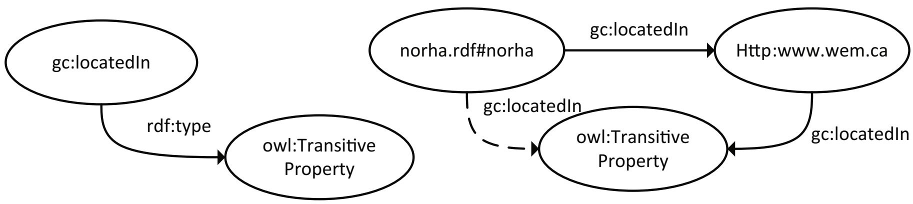

Fig. 9. Inferring implicit context facts with Rule 7

Rule 7 enables transitivity for any property that is defined as a transitive property in SMARTERCONTEXT. In the example, since `gc:locatedIn` is a transitive property, Norha is located in West Edmonton Mall (`http://www.wem.ca`), and this mall is located in Edmonton, SMARTERCONTEXT infers from Norha's context sphere that she is located in Edmonton. Location-based context facts are crucial to suggest user-centric deals, products, and services effectively.

Reasoning from Symmetric Properties. The following deduction rules exploit the semantic characteristics of symmetric properties in OWL-Lite.

Rule 8 $(?A\, rdf : Type\; owl : SymmetricProperty), (?u\; ?A\; ?v) \rightarrow (?v\; ?A\; ?u)$

Example:

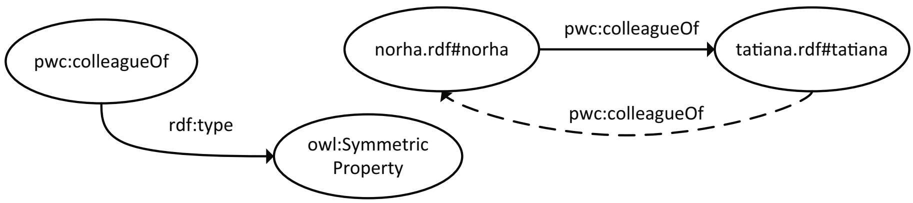

Fig. 10. Inferring implicit context facts with Rule 8

Symmetric properties state that if the context relationship represented by the property is valid for subject u and object v, it is valid also for v acting as the subject and u as the object of the relationship. `pwc:colleagueOf` is a symmetric property in SMARTERCONTEXT. Therefore, given that from Norha's context sphere the human entity `tatiana.rdf#tatiana` is a colleague of Norha, Norha is a colleague of Tatiana.

Reasoning from Inverse Properties. The following deduction rules exploit the semantic characteristics of the `owl:inverseOf` object property.

Rule 9 $(?A\; owl : inverseOf\; ?B), (?u\; ?A\; ?v) \rightarrow (?v\; ?B\; ?u)$

An interesting application of inverse properties in smarter commerce and in general in the smart internet is the inference of social relationships between context entities. `pwc:parentOf` and `pwc:childOf` are examples of object properties that are inverse to each other in the SMARTERCONTEXT ontology. For example (cf. Fig. 11 below), given the context fact `(gabriel.rdf#gabriel pwc:parentOf jg.rdf#jg)`, it is possible to infer the fact `(jg.rdf#jg pwc:childOf gabriel.rdf#gabriel)`. Particularly in shopping scenarios, the shopping list of a parent could be affected by the shopping list of his kid and vice versa (although the second case is generally less probable than the first one).

Example:

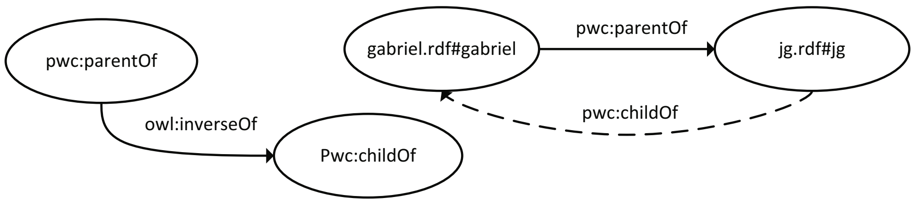

Fig. 11. Inferring implicit context facts with Rule 9

8.2 SMARTERCONTEXT Deduction Rules

This section presents selected rules that we defined in the SMARTERCONTEXT ontology to extend the standard reasoning capabilities provided by RDFS and OWL-Lite (cf. Section 8.1).

Rule 10 (pwc:NearTo) $(?a\ gc : locationRelationship\ ?b)$, $(?b\ pwc : isNearTo\ ?c) \rightarrow (?a\ pwc : isNearTo\ ?c)$

Example:

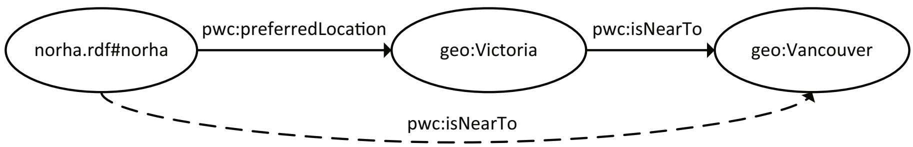

Fig. 12. Inferring implicit context facts with Rule 10

SMARTERCONTEXT uses Rule 10 to infer location-based context facts for the identification of relevant products, services, and retailers. This rule states that if an entity a is related to a location entity b by a `gc:locationRelationship`, and entity b is near to another location entity c, a valid conclusion is that entity a is near to c. By applying this rule to the example, given that Norha has Victoria as her preferred location, `pwc:preferredLocation` is a subproperty of `gc:locationRelationship`, and according to the `geo` vocabulary Victoria is near to Vancouver, it is possible to infer that Norha is near to Vancouver. As a result, Norha may be interested in products, services, and deals not only available in Victoria, but also in Vancouver.

Rule 11 (shopping:FamilyShoppingList) $(?a\ pwc : relativeOf\ ?b)$, $(?a\ shopping : toBuy\ ?c) \rightarrow (?b\ shopping : toBuy\ ?c)$

Example:

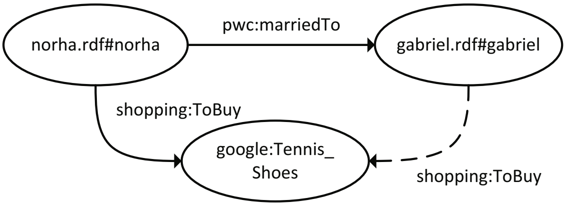

Fig. 13. Inferring implicit context facts with Rule 11

In our smarter commerce case study, Rule 11 is useful to infer products that could be included in the user's shopping list from the shopping lists of the user's relatives and vice versa. According to Norha's context sphere, tennis shoes and electric grills are product categories in her shopping list. Therefore, given that Norha is a relative of Gabriel, these two product categories could be suggested as Gabriel's shopping list products. Figure 11 depicts the application of the rule for tennis shoes.

Rule 12 (shopping:SocialBasedShoppingPreferences)
$(?a\ gc : socialRelationship\ ?b), (?a\ pwc : likes\ ?c) \rightarrow (?b\ pwc : likes\ ?c)$

Example:

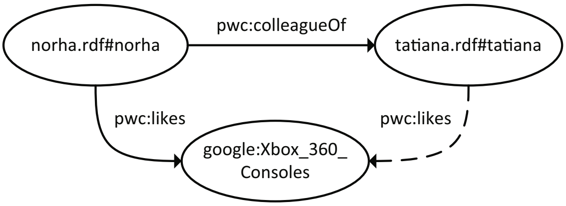

Fig. 14. Inferring implicit context facts with Rule 12

Rule 12 is comparable to Rule 11 but applies to more general social relationships (besides `pwc:relativeOf`) and to user interactions different than `shopping:toBuy` (i.e., `pwc:likes`, `pwc:dislikes`, and `pwc:isInterestedIn`). An example of the application of this rule involves the facts `(norha.rdf#norha pwc:likes google:XBox_360_Consoles)` and `norha.rdf#norha pwc:colleagueOf tatiana.rdf#tatiana`. Since Norha and Tatiana are colleagues and probably share interests and shopping preferences, it would be relevant to offer XBox 360 consoles or similar products to Tatiana.

Rule 13 (shopping:RelatedProductsPreferences)
$(?a\ shopping : relatedProductOrService\ ?b), (?c\ pwc : isInterestedIn\ ?a) \rightarrow (?c\ pwc : isInterestedIn\ ?b)$

Example:

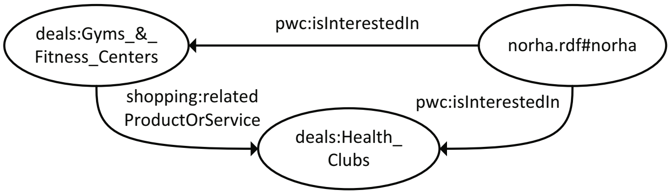

Fig. 15. Inferring implicit context facts with Rule 13

Relevant products, services and deals can be recommended by taking into account relevant products and services. Suitable ontologies for characterizing products, services and deals must support knowledge representation about related products (e.g., complementary products such as earrings and necklaces, or gyms and health clubs). In the example Norha is interested in deals related to gyms and fitness centers, therefore and taking into account that these deals are related to health clubs, she would be interested in deals related to the `deals:Health_Clubs` category.

9 Discussion and Related Work

Context modeling is an important component of the context information life cycle [22]. The smart internet and its applications such as the personal web require context models to represent the relevant aspects of entities that affect the interactions between users and systems, as well as the relationships between users and these entities. Ontologies are useful to describe concepts and the relationships among them. Therefore, ontology-based models are natural mechanisms to represent context information since context is a specific kind of knowledge [23]. The SMARTERCONTEXT ontology is a suitable mechanism for context representation and reasoning in the smart internet. It provides the mechanisms for the formal specification of the semantics of context data from a user-centric perspective [2]. Furthermore, an important modeling feature for realizing user-centric interactions and services in the smart Interent is knowledge sharing. The semantic web technologies supporting SMARTERCONTEXT not only allow the implementation of runtime context models, but also the interchange of context information among heterogeneous and distributed web entities.

Most context ontologies have been proposed for context representation and reasoning in pervasive and ubiquitous environments [2]. According to our systematic review of context modeling and management approaches, 49% of the surveyed approaches were proposed by the pervasive and ubiquitous computing research community. The remaining 51% are divided among several other communities: self-adaptive and self-organizing systems (12%), artificial intelligence and knowledge representation (11%), autonomic computing (8%), human computer interaction (5%), mobile computing and wireless networks (12%), and model-driven engineering (3%). To the best of our knowledge and according to relevant surveys on state-of-the-art context-aware computing, the SMARTERCONTEXT ontology is the first approach that has been proposed for user-centric context management in web applications [2,23,24].

Several ontologies are available for representing things in the semantic web. Examples of these ontologies are FOAF[10] (friend-of-a-friend), the ontology to connect people across the web [25]; GoodRelations,[11] the ontology for describing product and service offers on the web [26]; and GeoNames, the ontology that adds geospatial semantic information to the web.[12] In contrast to existing ontologies, SMARTERCONTEXT provides the common framework required by the smart internet to augment the semantics of existing ontologies to make them suitable for context representation. We envision SMARTERCONTEXT as the knowledge representation mechanism required to elevate the visibility of context information as demanded by smart web interactions and adaptive services. Nevertheless, without SMARTERCONTEXT, existing semantic web ontologies are useful to characterize and reason about web entities in concrete application domains with no awareness of user and system situations. For example, FOAF supports the representation of social relationships, GoodRelations the representation of online product and service offers, the Google Product taxonomy the representation of products, and GeoNames provides information about geographical places. The integration of these vocabularies into SMARTERCONTEXT instantaneously augments their semantics by converting their concepts into context entity types that can relate to each other to describe information about the user's situation. Therefore, these ontologies will represent not only web resources independent of the user, but relevant context about the user's situation. For example, FOAF would represent social context relationships that could be exploited to discover shopping preferences from the user's social network. GoodRelations and the Google Product taxonomy would describe not only products and services, but also the interactions between shoppers and online offers, thus enabling innovative approaches to leverage web interactions in business intelligence (BI) applications. Finally, the GeoNames ontology would represent not only places in the world, but geographical locations meaningful to improve the user's web experience.

10 Ongoing Research

Representing and managing context is not only critical for the realization of the smart internet and the personal web but also poses interesting research challenges. Our ongoing research concentrates on two of them: the management of trade-offs between expressiveness and performance, and the assurance of privacy and confidentiality of personal context data.

Performance is an important quality attribute to deliver user-centric smart interactions and services effectively. On the one hand, ontology-based knowledge representation approaches such as OWL expose performance limitations when reasoning on large data sets [8]. On the other hand, pure RDFS approaches lack semantic expressiveness

[10] http://www.foaf-project.org/

[11] http://www.heppnetz.de/projects/goodrelations

[12] http://www.geonames.org/ontology/documentation.html

for context reasoning [8,6,16]. An appropriate balance between expressiveness and performance is crucial to be able to reason about context situations with high amounts of context data. To balance this trade-off, we are investigating the application of computational biology algorithms and techniques to the mining of context facts [27,28]. Given the effective application of these techniques to the analysis of complex biological networks, we hypothesize that they can contribute to the analysis of complex RDF-based context models. These context mining techniques must be applied effectively not only to the analysis of individual context models, but also to the analysis of multiple context spheres (e.g., to correlate shopping preferences from personal context models of members of social networks). The development of suitable tools for the specification of context mining rules to be integrated into our context management engine at runtime complements this research.

To validate the SMARTERCONTEXT framework and the general applicability of the SMARTERCONTEXT ontology we are working on several case studies. Regarding the smarter commerce domain, we developed a deal recommendation system that exploits users' changing personal context information to deliver highly relevant offers. This application relies on recommendation algorithms based on collaborative filtering, and SMARTERCONTEXT. SMARTERCONTEXT provides the deal application with up-to-date information about user locations and product preferences gathered from their past and present web interactions. We conducted several experiments using real datasets to simulate personal context information gathered by SMARTERCONTEXT. For many deal categories the accuracy of the solution enhanced with SMARTERCONTEXT was between 3% and 8% better than the approaches we used as baselines. For some categories, and in terms of multiplicative relative performance, it outperformed related approaches by as much as 173.4%, and 37.5% on average [20].

Another relevant application to validate our research on context-awareness is the management of service level agreements (SLAs) in SOA governance [17]. The cornerstone of the SOA governance case study is the realization of *context-driven* SLAs, an extension of SLAs where context monitoring requirements are explicitly mapped to quality of service objectives to optimize the runtime control of contracted obligations. In our SOA case study, we extended the SMARTERCONTEXT ontology from the GC taxonomy to define the context types, and context relationships required to model, from SLA specifications, context monitoring requirements and context management strategies that change at runtime. Therefore, RDF graphs represent both context information and context management strategies (i.e., context gatherers and monitoring conditions).

A third case study we are conducting is the application of SMARTERCONTEXT to the monitoring of adaptation properties and goals for supporting runtime V&V of self-adaptive software [29]. In this research, the SMARTERCONTEXT ontology supports the representation of context monitoring requirements derived from adaptation properties and goals. For this, we mapped SMARTERCONTEXT types to the adaptation properties proposed in our evaluation framework for quality-driven self-adaptive systems [30]. Since adaptation goals can evolve over time, SMARTERCONTEXT supports dynamic changes in the monitoring infrastructure to preserve the relevance of monitoring strategies with the situation of the adaptive system.

11 Conclusions

Context awareness is a fundamental requirement to support the discovery, aggregation, and delivery of services according to user preferences and situations. Therefore, the effectiveness of smart interactions and services depends on the suitability of context representation and context reasoning techniques to understand the situation of users and systems. This paper explained the SMARTERCONTEXT ontology and its application to the smart internet using a personal web case study in smarter commerce. The SMARTERCONTEXT ontology exploits the semantic web to leverage context-awareness and thus optimize the user's experience in the smart internet. It provides a foundational framework to integrate existing semantic web vocabularies. This integration is crucial for raising the visibility of context information in user-centric, context-aware web applications. SMARTERCONTEXT augments the semantics of web resources represented by existing vocabularies. As a result, web resources with which the user interacts evolve from "things" in the web that are disconnected from the user into meaningful context entities that are now crucial for the understanding of personal preferences and situations. Most importantly, since SMARTERCONTEXT allows the understanding of the interactions between the user and web systems, web interactions evolve from simple data input mechanisms into the means to discover relevant context entities.

Finally, the formalization of a user model of the web centered on users and their goals constitutes one of the two main research challenges stated in the smart internet's research agenda [1]. Such a model must include the specification of user preferences and situations explicitly, and must provide runtime support for the manipulation of this context information. The SMARTERCONTEXT ontology provides the basis for context knowledge representation in the user-centered model of the web required by the smart internet.

Acknowledgments. This work was funded in part by the National Sciences and Engineering Research Council (NSERC) of Canada under the NSERC Strategic Research Network for Smart Applications on Virtual Infrastructures[13] (SAVI - NETGP 397724-10) and Collaborative Research and Development program (CRDPJ 320529-04 and CRDPJ 356154-07), IBM Corporation, University of Victoria (Canada), and Icesi University (Colombia).

References

1. Ng, J.W., Chignell, M., Cordy, J.R., Yesha, Y.: Overview of the Smart Internet. In: Chignell, M., Cordy, J., Ng, J., Yesha, Y. (eds.) The Smart Internet. LNCS, vol. 6400, pp. 49–56. Springer, Heidelberg (2010)
2. Villegas, N.M., Müller, H.A.: Managing Dynamic Context to Optimize Smart Interactions and Services. In: Chignell, M., Cordy, J., Ng, J., Yesha, Y. (eds.) The Smart Internet. LNCS, vol. 6400, pp. 289–318. Springer, Heidelberg (2010)
3. Coutaz, J., Crowley, J.L., Dobson, S.: Context is Key. Communications of the ACM (CACM) 48(3), 49–53 (2005)

[13] http://www.nsercpartnerships.ca/How-Comment/Networks-Reseaux/SAVI_AIIV-eng.asp

4. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data — The Story So Far. International Journal on Semantic Web and Information Systems 5(3), 1–22 (2009)
5. Villegas, N.M., Müller, H.A., Muñoz, J.C., Lau, A., Ng, J., Brealey, C.: A Dynamic Context Management Infrastructure for Supporting User-driven Web Integration in the Personal Web. In: 2011 Conference of the Center for Advanced Studies on Collaborative Research (CASCON 2011), pp. 200–214. IBM Corp., Markham (2011)
6. Manola, F., Miller, E.: RDF Primer. Technical report, W3C (2004)
7. Munoz, J.C., Tamura, G., Villegas, N.M., Müller, H.A.: Surprise: User-controlled Granular Privacy and Security for Personal Data in SmarterContext. In: Proceedings 2012 Conference of the Center for Advanced Studies on Collaborative Research (CASCON 2012), pp. 131–145. IBM Corp., Riverton (2012)
8. Hitzler, P., Krötzsch, M., Rudolph, S.: Foundations of Semantic Web Technologies, 1st edn. Textbooks in Computing, vol. 33. Chapman & Hall/CRC (2009)
9. Berners-Lee, T., Hall, W., Hendler, J.A., O'Hara, K., Shadbolt, N., Weitzner, D.J.: A Framework for Web Science. Foundations and Trends in Web Science 1(1), 1–130 (2006)
10. The World Wide Web Consortium (W3C): RDF Vocabulary Description Language 1.0: RDF Schema (2004), `http://www.w3.org/TR/rdf-schema/`
11. The World Wide Web Consortium (W3C): OWL Web Ontology Language Reference (2004), `http://www.w3.org/TR/owl-ref/`
12. Berners-Lee, T., Fielding, R., Masinter, L.: Uniform Resource Identifier (URI): Generic Syntax (January 2005), `http://www.ietf.org/rfc/rfc3986.txt`
13. Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T.: Hypertext Transfer Protocol - HTTP/1.1 (1999), `http://www.w3.org/Protocols/rfc2616/rfc2616.html`
14. Knublauch, H., Fergerson, R.W., Noy, N.F., Musen, M.A.: The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 229–243. Springer, Heidelberg (2004)
15. The World Wide Web Consortium (W3C): XML Schema Part 2: Datatypes - W3C Recommendation. (May 2001), `http://www.w3.org/TR/xmlschema-2/`
16. The World Wide Web Consortium (W3C): OWL Web Ontology Language Overview (2004), `http://www.w3.org/TR/2004/REC-owl-features-20040210/#s3.1`
17. Villegas, N.M., Müller, H.A., Tamura, G.: Optimizing Run-Time SOA Governance through Context-Driven SLAs and Dynamic Monitoring. In: 2011 IEEE International Workshop on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA 2011), pp. 1–10. IEEE (2011)
18. Google Inc.: The Google Product Taxonomy (2012), `http://support.google.com/merchants/bin/answer.py?hl=en&answer=160081`
19. Groupon: The Groupon Deal Categories (2012), `https://sites.google.com/site/grouponapiv2/api-resources/deals`
20. Ebrahimi, S., Villegas, N.M., Müller, H.A., Thomo, A.: SmarterDeals: A Context-aware Deal Recommendation System based on the SmarterContext Engine. In: Proceedings 2012 Conference of the Center for Advanced Studies on Collaborative Research (CASCON 2012), pp. 116–130. IBM Corp., Riverton (2012)
21. Carroll, J.J., Dickinson, I., Dollin, C., Seaborne, A., Wilkinson, K., Reynolds, D.: Jena: Implementing the Semantic Web Recommendations. In: 13th International World Wide Web Conference (WWW 2004), pp. 74–83 (2004)

22. Hynes, G., Reynolds, V., Hauswirth, M.: A Context Lifecycle for Web-Based Context Management Services. In: Barnaghi, P., Moessner, K., Presser, M., Meissner, S. (eds.) EuroSSC 2009. LNCS, vol. 5741, pp. 51–65. Springer, Heidelberg (2009)
23. Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., Riboni, D.: A Survey of Context Modelling and Reasoning Techniques. Pervasive and Mobile Computing 6, 161–180 (2009)
24. Hoareau, C.: Modeling and Processing Information for Context-Aware Computing: A Survey. New Generation Computing 27(3), 177–196 (2009)
25. Graves, M., Constabaris, A., Brickley, D.: FOAF: Connecting People on the Semantic Web. Cataloging & Classification Quarterly 43(3-4), 191–202 (2007)
26. Hepp, M.: GoodRelations: An Ontology for Describing Products and Services Offers on the Web. In: Gangemi, A., Euzenat, J. (eds.) EKAW 2008. LNCS (LNAI), vol. 5268, pp. 329–346. Springer, Heidelberg (2008)
27. Pavlopoulos, G., Secrier, M., Moschopoulos, C., Soldatos, T., Kossida, S., Aerts, J., Schneider, R., Bagos, P.: Using Graph Theory to Analyze Biological Networks. BioData Mining 4(1), 1–27 (2011)
28. Milo, R., Al, E., Biology, C.: Network Motifs: Simple Building Blocks of Complex Networks. Science, 824–827 (2002)
29. Tamura, G., Villegas, N.M., Müller, H.A., Sousa, J.P., Becker, B., Karsai, G., Mankovskii, S., Pezzè, M., Schäfer, W., Tahvildari, L., Wong, K.: Towards Practical Runtime Verification and Validation of Self-Adaptive Software Systems. In: de Lemos, R., Giese, H., Müller, H.A., Shaw, M. (eds.) Software Engineering for Self-Adaptive Systems. LNCS, vol. 7475, pp. 108–132. Springer, Heidelberg (2013)
30. Villegas, N.M., Müller, H.A., Tamura, G., Duchien, L., Casallas, R.: A Framework for Evaluating Quality-driven Self-Adaptive Software Systems. In: 6th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2011), pp. 80–89. ACM, New York (2011)