Exploiting Social Context in Personalized Web-Tasking Applications

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

[1]University of Victoria, Victoria, Canada
[2]Icesi University, Cali, Colombia
[3]IBM Canada Software Laboratory, CAS Research. Markham, Canada

Abstract

Personalized Web-Tasking (PWT) systems automate ordinary and repetitive web interactions while exploiting personal context to deliver personalized features. Among the personal context of a user, *social context* is all information obtained from the relationships with other users, which is relevant to the user's personalized web-tasks. Current approaches exploit the information of social media, or the explicit input of the user, and use it as is. In addition to this, PWT systems also benefit by inferring social relationships through reasoning over such information and other sources of context. For example, a calendar application might record events the user shares with other people, or the sensors on mobile devices can be used to identify others nearby. This information can be exploited to improve the execution of PWT applications including its personalization and context-adaptive capabilities. In this paper, we present our ongoing research and implementation to enable PWT systems with capabilities to exploit social context dynamically: (i) our extension of the PWT Ontology, and (ii) our SMARTERCONTEXT inference rules approach to select relevant social context for PWT.

1 Introduction

Personalized Web-Tasking (PWT) is the automation of repetitive and mundane web interactions—that are associated with a personal goal of a user—by taking advantage of all relevant context to deliver personalized functionalities and improve his or her web experience [4, 18].

Context-awareness is indispensable for realizing PWT applications. That is, to understand the execution environment as well as the situations and web goals of the users. Context-awareness in PWT systems facilitates two main objectives [5]: (1) guarantee self-adaptive capabilities in order to perform adaptations over itself under changing contextual conditions [7, 17, 12]; and (2) exploit personal context (i.e., the user's information, web interactions, and social sphere) to provide better levels of personalization when executing web-tasks on behalf of the user. In this paper, we focus on context-awareness based on the personal context gathered from the social relationships of the user with other users—*social context*.

There are various approaches for exploiting social context in different domains [2, 20, 13, 15]. For example, the AWARE architecture aims to improve awareness in a hospital setting. The main feature incorporates information about other users (physicians and nurses) into the system and thereby improves its functionalities [2].

Perhaps the most common way of integrating social context is by using the user's social media platforms (e.g., Facebook, Twitter, or LinkedIn). Social media exposes plenty of personal information about the user (e.g., software networks that describe preferences, personal data, generated contents, and social connections). However, some web interactions and applications remain as unexploited sources of social context. For instance, (i) calendar applications that record special dates and events (e.g., birthdays, anniversaries, graduations, or meetings); (ii) web activities related to communities of interest or practice that enrich the user's social sphere (e.g., blogs, forums, collaboration projects, or peers communication); and (iii) other personalized web-tasking context such as the information from other users' personal goals. PWT applications also benefit from implicit social context, which requires proper instrumentation to represent, monitor, and reason about it. Not to mention, sensors and infrastructures to acquire such information.

Personalized web-tasking involves multiple services and systems that are frequently generating context information. Given that context-awareness is a key feature in PWT applications, we discuss some relevant approaches for representing social context using taxonomies and ontologies [23, 19, 3], and reasoning about it using rules and context patterns [3, 25]. More importantly, we present an extension of our PWT ontology to represent social context and the impact on PWT systems, and propose an extension for the SMARTERCONTEXT Reasoning Engine inference mechanism proposed by Villegas [24]. This extension comprises a set of rules to be selected at runtime, for managing relevant social context for PWT.

The remainder of this paper is organized as follows: Section 2 discusses relevant approaches for representing and managing social context. Section 3 describes our PWT system approach and the social context influence through a scenario related to an "attending a conference" personal goal. Section 4 describes the SMARTERCONTEXT framework, which we exploit to provide context-awareness capabilities for PWT systems based on our social context inference rules. Section 5 presents our social context modelling approach for PWT as an extension of our PWT Ontology. Section 6 describes our inference rules to decide upon relevant social context based on context facts. Finally, Section 7 concludes this paper with a summary and a discussion of future work.

2 Relevant Approaches

Personalized web-tasking requires personal context information to deliver user-centric features. For the past decade there have been numerous approaches that exploit user context to deliver web automation and personalization for different aspects of the web including, content (i.e., deciding what information is interesting for the user), presentation (i.e., how the information should be displayed), and navigation (i.e., what sequence of web sites the user should follow) [1, 8, 6, 9, 14, 11, 22].

In addition, the *web of things* as an ecosystem of devices, applications and people [21] has turned context exploitation a relevant feature for web and mobile applications. However, the exploitation of social context had been mostly limited to certain sources of information such as those related with social networks and mobile sensors [20, 15, 2]. For instance, Lukowicz et al. argue that smartphones have all sensing capabilities to exploit context and propose that software systems move to a scenario in which the user interacts with communities of users with the goal of accomplishing complex tasks collaboratively [13]

In addition to what has been achieved on these approaches, different techniques for context monitoring, acquisition, modelling and reasoning [25] are the main features in the implementation of PWT. This section summarizes relevant approaches in conceptualizing, representing, and managing social context.

2.1 Social Context Representation

Many of the current applications (web and mobile) include social media features (e.g., login, share or like) with the purpose of exploiting the users' social connections as part of the application's functionality. Perhaps the most popular social network representation is the Facebook social graph—also known as the Open Graph API[1]—which represents the social relationships of the user in a social network. This graph connects a user with other elements (e.g., users, photos, and events) through specific connections (e.g., friendship, tags, and shared content). More importantly, it represents the interests of the users in terms of *likes* over these elements.

However, social networks are not the only source of social context. Moreover, its representations have been approached by different research groups. For instance, the *So Smart* framework proposed by Biamino [3], for modelling social context, exploits all information in the user's social network

[1] https://developers.facebook.com/docs/opengraph

to empower smart objects to understand social contexts, thus improving user activities. This framework uses ontology-based context modelling to provide reasoning capabilities. Their approach defines three ontologies: (i) object, to represent the object and provide self-awareness; (ii) social context, to represent the connections of the object with others; and (iii) social goal, to represent the actions and objects that play a role in the achievement of a specific goal—individually in a group. Also, their definition of social context is based on the structure of the user's social network as a 3-tuple of: size, density, and type of connections. In addition, their social context ontology includes typical classifications into groups borrowed from sociology such as tribe, community, family, peer, club, and household.

Smart devices are instrumented with sensors capable of gathering information about the user, hence social context modelling has also been addressed by the domain of pervasive computing. The *pervasive social context* of a user is defined by Schuster et al. [23] as all information generated by the user's social interactions with other people—directly or indirectly—while using devices equipped with respective sensors to gather such information from the user's social network services (e.g., profile information, activities, and other people in the user's social graph).

Schuster et al. proposed a taxonomy—called STiPI—for pervasive social context, based on four dimensions: Space, Time, People, and Information Source. Table 1 summarizes the STiPI taxonomy [23]. The *Space* dimension determines physically *where* the social interactions are consuming or producing context information. The *Time* dimension determines *when* the social interaction happens, and the *People* dimension the *who* are the individuals involved. Finally, the *Information source* dimension determines *how* the social context information is generated. That is, from a physical or virtual source (e.g., device or social network), or the composition and integration of both.

Finally, the Social Ontology and Semantic Actions (SOSA) proposed by Ng and Lau is a social networking model that seeks to consolidate a variety of contacts from the user's multiple social networks [19]. For this purpose, SOSA provides an ontology to manage relationships, and semantic actions to integrate these relationships with enterprise services. SOSA's conceptual design includes: (i) a social input source, through which the user can select which of her social networking platforms she wishes to integrate; (ii) the *contact graph object* that comprises links among the contacts and the different social platforms; (iii) the *relationship axiom* that expresses a relationship of the user with a contact, and an extensible *relationship ontology* that defines the set of relationships; and (iv) the *semantic actions template* that associates semantic actions to a relationship axiom.

Table 1: A summary of the STiPI Taxonomy [23].

Dimension	**Levels**
(S)pace	1. Small 2. Medium 3. Anywhere
(Ti)me	1. Short-term 2. Mid-term 3. Long-term
(P)eople	1. Individuals 2. Groups 3. Anonymous, Community
(I)nformation Source	1. Pervasive sensors 2. Social networks 3. Sources integration

Although these approaches provide information about the user's social connections, the realization of PWT as self-adaptive systems requires models that evolve at runtime and can be manage dynamically.

2.2 Social Context Monitoring and Reasoning

Self-adaptive software systems imply context-aware support [12]. That is, monitoring and reasoning about context are key features in their implementation. Reasoning about context requires infrastructure to manipulate the representation of context and provide the system with relevant information for its operation.

Biamino [3] proposed their reasoning approach based on rules applied over a social graph derived from their ontologies. They implement three reasoning mechanisms: network-driven (i.e., examining the information gathered from their social network), group-driven (i.e., using inferences from the behaviour of a group rather than the individual itself), and learning (i.e., acquired knowledge to increase the objects' intelligence).

In the implementation of PWT applications, we exploit the SMARTERCONTEXT Monitoring Infrastructure and Reasoning Engine proposed by Villegas [24] to enable context-aware capabilities into the system [5]. The SMARTERCONTEXT Reasoning Engine (SCoRE) uses sensors to monitor context and then applies inference rules or context patterns to discover implicit context facts from explicit context information.

3 PWT System Overview

We propose to realize PWT applications as situation-aware self-adaptive software systems, which involves three concerns: (i) proper models (to represent the system and other relevant elements, as well as their interactions) operable and evolvable at runtime; (ii) context-awareness to understand and reason about changes in the users' situations and the environment; and (iii) self-adaptive support to adapt itself upon these changes.

We propose two runtime models for PWT systems: our PWT model, and our goal-oriented context-sensitive web-tasking (GCT) model [5]. The first one is an ontology, available as an OWL/RDF specification,[2] that defines the conceptual elements of PWT (i.e., personal goal, observable result, satisfaction property, web-task sequence, web sub-task, activity, operation, and information resources), and the relationships among them (i.e., dependency, causality, and sequencing). A simplified view of our PWT ontology is depicted in Figure 1. The second one allows the modelling of the relevant context, the personal goals and subgoals, and the details of the associated web subtasks (i.e., services, inputs, and outputs). This model uses our extension of the iStar framework elements [10] tailored to the personalized web-tasking domain.

These runtime models are processed by our self-adaptive PWT system depicted in Figure 2, which comprises four software components [5]:

(1) *Web-Tasking Knowledge Infrastructure:* this component allows users to express personal goals explicitly (the user specifies a goal to be fulfilled) or implicitly (the system learns from the user's behaviour). Moreover, the web interactions associated with the fulfilment of the personal goal are recorded (manually or automatically) as instances of our GCT model.

(2) *PWT Model Processor:* this component is responsible for analysing and transforming GCT instances into RDF graphs that contain the specific user web-tasking associated with a personal goal.

(3) *Personalization Engine:* this component exploits user context to provide personalized features during the web-tasking execution.

(4) *Web-Tasking Effector:* this component is responsible to executes and manage the personalized web tasks on behalf of the user, as well as to evaluate if the personal goal has fulfilled.

Additionally, we exploit the SMARTERCONTEXT monitoring infrastructure to guarantee context-awareness support, and the SMARTERCONTEXT Reasoning Engine (SCoRE) to provide logic capabilities over the context.

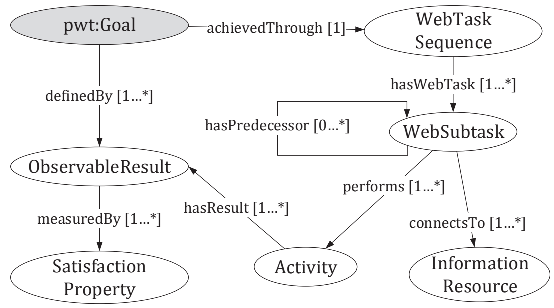

Figure 1: Simplified view of our PWT ontology model [5]

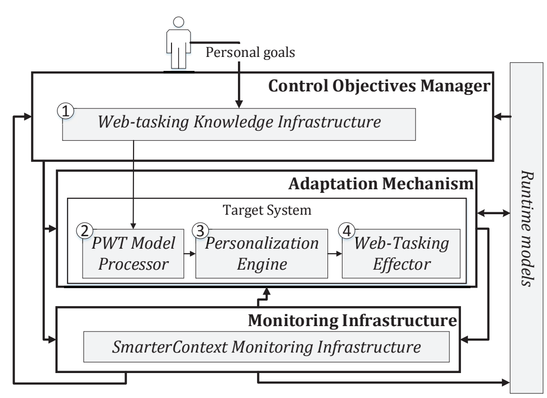

Figure 2: Out PWT system overview [5]

[2]http://www.rigiresearch.com/research/pwt/pwtOntology.owl

3.1 Case Study: "Attending a Conference" Personal Goal

To illustrate the relevance of social context in PWT we use the following scenario: Lorena is a student at the University of Victoria and will be attending the CASCON conference located in Markham from November 3rd to the 5th. To fulfil her personal goal of attending this conference she will be performing web tasks associated with three subgoals (sg):

sg-1 Registering for the conference.

sg-2 Booking transportation from her current location to the conference city.

sg-3 Booking accommodation for the dates of the conference.

Each of these subgoals comprises an independent sequence of web-subtasks that are instantiated from our GCT model and managed by our PWT system. For instance, *book transportation* in our example implies six web-subtasks (wst) for Lorena (cf. Figure 3):

wst-1 Find the proper airline based on preferences and airline's coverage.

wst-2 Apply specific filters such as departure and arrival cities and dates.

wst-3 Select a flight based on personal preferences such as the number of stops and seat availability, and other conditions such as budget price.

wst-4 Input information about the passenger and other flight details.

wst-5 Purchase the ticket.

wst-6 Create a calendar event with the flight details with a reminder—three days prior departure.

The PWT system executions generate context facts about the user represented by elements from our PWT Ontology and the SMARTERCONTEXT Ontology. For instance, Figure 4 depicts our scenario showing three context facts about Lorena: (1) an association with a *pwt:Event* element named *CASCON Conference* through the object property *pwt:attendsTo* that; (2) an association with a *pwt:Transport* element named *Flight for CASCON*, through an object propery: *pwt:hasTransportation*; and (3) an association with a the *pwt:Accommodation* element named *Hotel at CASCON* through the object property *pwt:hasAccommodation*.

4 The SmarterContext Framework

The SMARTERCONTEXT framework proposed by Villegas [24] comprises (1) the SMARTERCONTEXT Ontology to represent context, which is composed of the foundational ontology *GeneralContext (gc)*, and two domain specific extensions. One of these extensions, the *Personal* ontology, is relevant to user-centric applications as PWT Systems; and (2) the Reasoning Engine (SCoRE) that infers contextual facts from explicit context representations compliant with the SMARTERCONTEXT ontology.

SMARTERCONTEXT can support PWT systems. For example, the *Personal* ontology[3] defines the *socialRelationship* object property that is used to express social connections of the user. This property defines a relationship of a user with other user (or group of users) by specifying the type of relational context. For instance, *affiliatedWith, colleagueOf, friendOf*, and *relativeOf* are examples of sub-properties of *socialRelationship* to describe these relationships. Also, the object property *locationRelationship* defined in the gc ontology, is useful for example to identify nearby users affect the execution of the user's PWT (e.g., looking for a place to have dinner might include food preferences of those nearby users assuming they are all going together). Furthermore, the SCoRE's rules and patterns provide support to infer social context facts from explicit context information. Our PWT systems exploit this capability to discover social context.

The SCoRE functionality is based on the implementation of *contextual RDF graphs* which are directed graphs, whose vertices and arcs correspond to types defined in the SMARTERCONTEXT ontology including its extensions. A contextual RDF graph is defined as a set of *contextual RDF triples* (i.e., string representations that comprise a *subject*, a *predicate*, and an *object*) that define contextual facts or observations. For instance, to represent a relationship between two users—colleagues

[3] http://smartercontext.org/ontologies/personal/v0.1/personal.owl

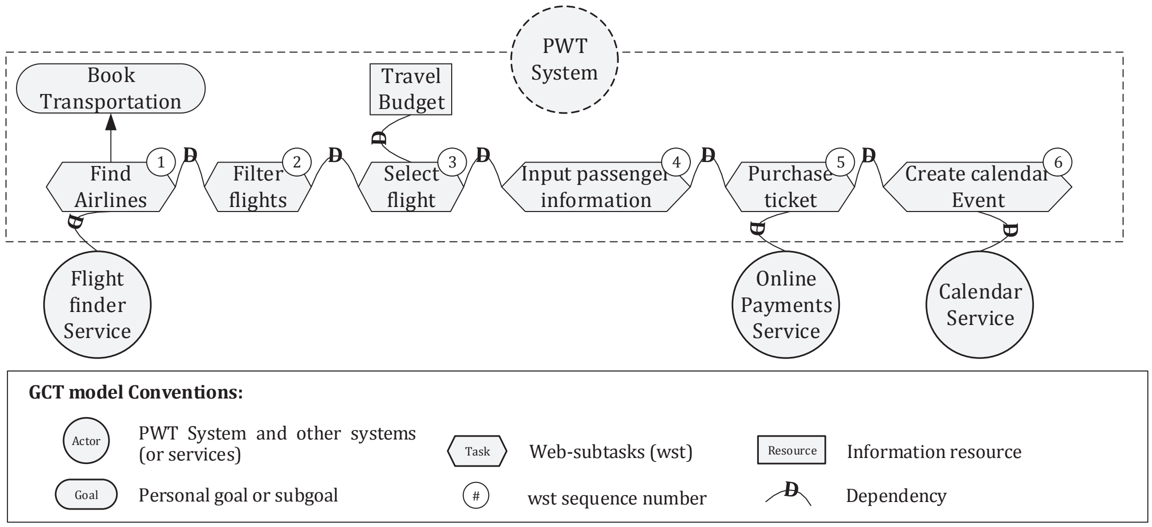

Figure 3: GCT model instance for Lorena's *book transportation* subgoal web-tasking

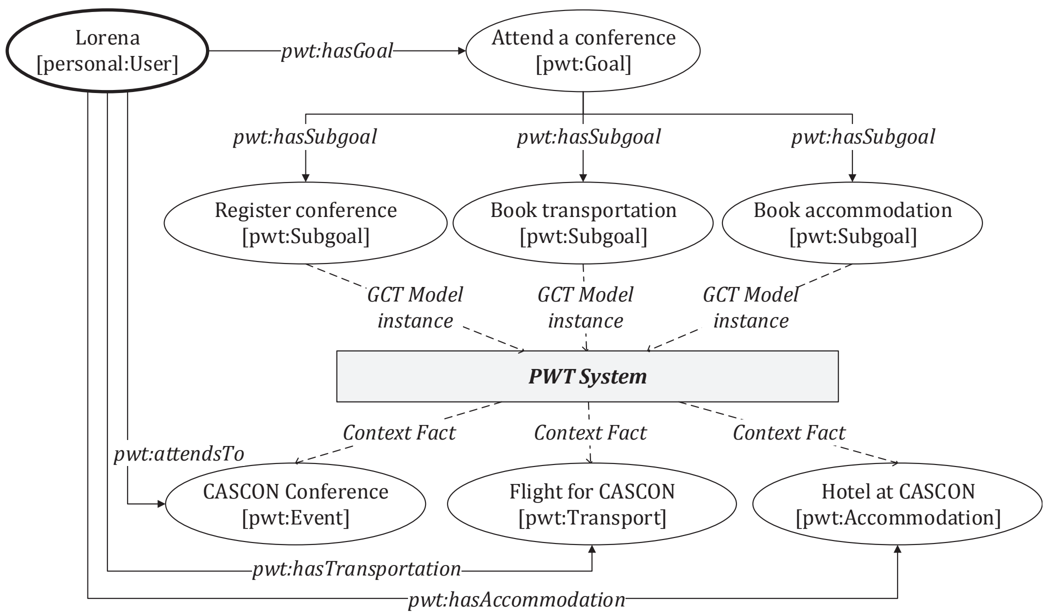

Figure 4: PWT scenario for Lorena's personal goal of "attending a conference"

Lorena and Nina—the contextual RDF triple would be *(Lorena, colleagueOf, Nina)*. In this contextual RDF triple, both subject (Lorena) and object (Nina) are context entities derived from the type *gc:HumanEntity* of the SMARTERCONTEXT. The predicate is an object property whose domain and range is *HumanEntity*.

The foundational reasoning mechanism of SCoRE is based on inference rules supported by the RDF Schema (RDFS) and OWL, including user-defined rules based on these two specifications. These rules are implemented in the Apache JENA semantic web platform.[4] A rule is composed by at least two triples connected by logic operators. The subject and the object of a triple may correspond to a variable. Variables are denoted using question

[4] http://jena.sourceforge.net/inference

marks. Predicates correspond to the ones defined in the SMARTERCONTEXT.

For example, the following rule allows the inference of a collegiality relationship through the concept of *association*.

Colleagues Rule:
(?A personal:associates ?B),
(?A personal:associates ?C)
→ *(?B personal:colleagueOf C?).*

For example, for two users Lorena and Andi that are associates of the Rigi research group the rule is: (Rigi personal:associates Lorena), (Rigi personal:associates Andi) → (Lorena personal:colleagueOf Andi).

The second inference mechanism of SCoRE is based on structural *context patterns* defined as *contextual RDF subgraphs* that provide a template to specify domain-specific context reasoning rules. There are different patterns in SCoRE proposed by Villegas [24] including: transitive, symmetry, inverse, join, generalization and delegation.

These patterns provide more time-efficient reasoning mechanisms in comparison with inference rules. The application of the symmetry pattern to our example allows us to infer that because Lorena is a colleague of Andi (explicit context) then Andi is a colleague of Lorena as well (implicit context).

The SMARTERCONTEXT engine is ideally applicable to the implementation of PWT systems. Not only it does define a flexible context ontology that can be extended with concepts of PWT domains, but also provides a reasoning mechanism to infer contextual information as required for example on the exploitation of social context for PWT systems.

5 Social Context for PWT

We define *social context for personalized web-tasking* as any information gathered from the social interactions of the user that is relevant to the users' personal goal.

In order to model and understand social context in PWT, we have added two entity types to our PWT Ontology (cf. Figure 5) that allows us to socially connect users during PWT executions:

- *pwt:Situation*: is defined as any user event that can be described in terms of time and location dimensions (cf. Figure 5). In PWT a personal goal is associated with a situation and viceversa. For example, in our scenario the associated situation is the "CASCON 2014 conference" located in "Markham" during the dates of "November 3rd to the 5th". The dimensions of a PWT Situation are represented as: (i) time by two data properties *validFrom* and *validTo* from the PWT ontology, and (ii) location by the object property *gc:locatedIn* and the *LocationContext* of the SMARTERCONTEXT *GeneralContext* ontology.
- *pwt:Resource*: a virtual element (e.g., service, application, or file) described in terms of a location dimension. In PWT if a web-subtask requires access to a resource, then the personal goal that is achieved using such web-subtask requires access to that resource as well. In out case study, the accessed resource is the "travel budget file" Lorena requires during her "selecting a flight" web-subtask. Likewise, the location dimension of the *pwt:Resource* is represented using the *LocationContext* of the SMARTERCONTEXT *GeneralContext* ontology through the object property *gc:locatedIn*.

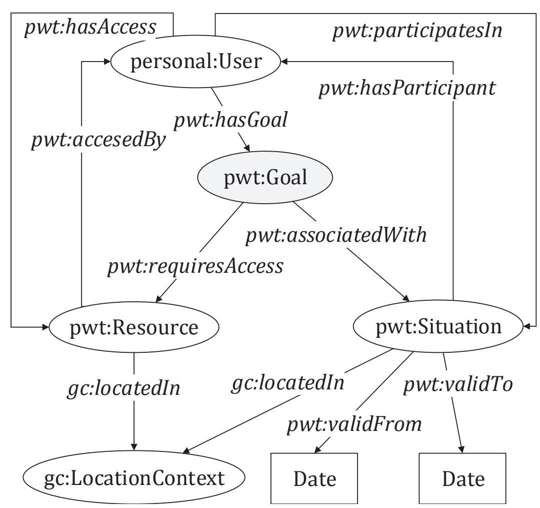

Figure 5: New entity classes added to our PWT Ontology: Situation and Resource

As mentioned before in our case study, Lorena has the personal goal "attending a conference" in Markham from November 3rd to the 5th. Hence, the personal goal is associated with a situation that is described in temrs of both location and time dimensions. Using the pwt:Situation entity type we can instantiate this conference event and associate it with Lorena as depicted in Figure 6.

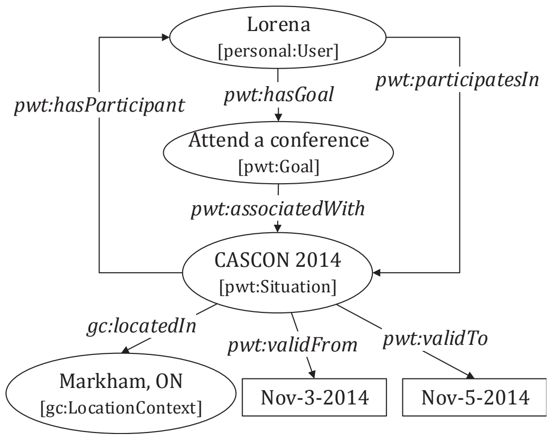

Figure 6: PWT Situation of our example for attending a conference

To illustrate how these new elements provide social context awareness for PWT applications, lets assume that Nina (a colleague of Lorena) wants to attend CASCON and is executing her own PWT web-task sequences to achieve the same "attending a conference" goal of Lorena. Because Nina's goal is associated with the same situation in which Lorena is participating, and because there is an explicit social connection between them (i.e., Nina has on her personal context that Lorena is a colleague of her), the *personalization engine* component in the PWT system can exploit Lorena's PWT information to enhance Nina's own PWT. For instance, Figure 7 depicts Lorena's PWT Transport details (i.e., her flight information: airline, departure time, and price) that are taken into consideration by the personalization engine component.

It is worth mentioning that social context allows us to determine whether two or more uses are socially relevant (i.e., there is a social connection between them or the users are similar). If they are relevant, each user's PWT system will exploit GCT instances of each other. For instance, if Lorena has a different web-subtask than Nina, when fulfilling the *book transportation* subgoal in which Lorena visits a website that provides a comparison of all possible airlines sorted by price; Nina's PWT system (specifically the personalization engine) might consider this new web-subtask as relevant for Nina

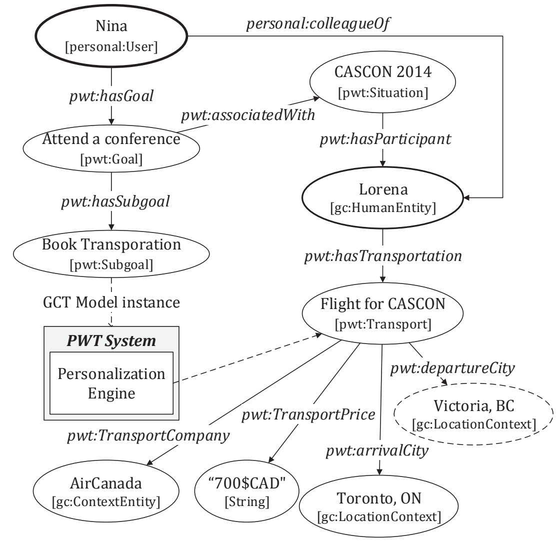

Figure 7: Lorena's flight information is considered during Nina's PWT of booking transportation

and include it in her PWT sequence.

Nevertheless, deciding on the relevance and reasoning of social context for a particular personal goal is not a trivial task. Next section explains how we address this challenge.

6 Social Context Reasoning for PWT

In our approach, we classify context in three categories:

(a) Explicit: represented by contextual facts included explicitly as RDF truples in the user's personal context. For example, Figure 8a shows explicit context: Nina is a colleague of Lorena.

(b) Implicit: inferred from expicit context facts. For example, Figure 8b depicts explicit context: Lorena and Andi are associated with Rigi research group From these facts and by specific rules, it can be inferred that Lorena is a colleague of Andi. Another example is to apply the SCoRE symmetry pattern [24] applied to the triple of Figure 8a to infer that Lorena is a colleague of Nina.

(c) Sensed: gathered from the physical and virtual world. For instance, the context fact depicted in Figure 8c could represent location context gathered through Lorena's and Stephan's mobile devices.

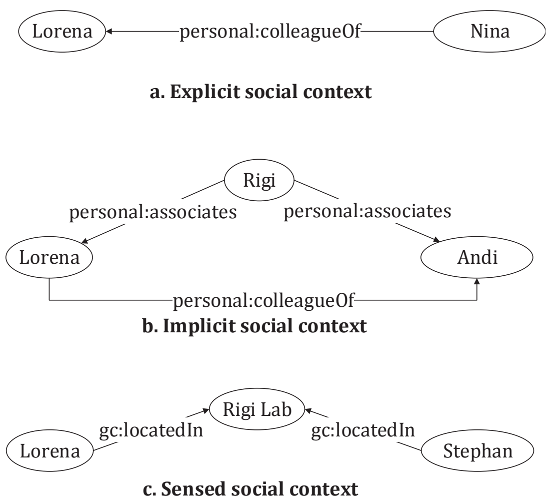

Figure 8: Categories of context depending on the source

As mentioned before, the SMARTER-CONTEXT Ontology contains elements to represent social context (i.e., the object properties *gc:socialRelationship* and *gc:locationRelationship*), and the SCoRE inference mechanisms (i.e., rules and context patterns) provide support to infer social relationships among users. However, the personalization engine in our PWT system—which is responsible for exploiting context to improve PWT—requires the SCoRE to reason upon the relevance of social context. That is, to decide which of all the social connections of the user are in fact relevant and can be exploited (e.g., by including context facts or other users GCT instances) during the execution of the user's PWT.

In our approach, *social context relevance* implies that a social relationship between the users does exist (i.e., explicit, inferred or sensed) in the personal context of the user; or the users are similar. Users similarity can be determined using data analysis techniques (e.g., collaborative filtering).

To enable our PWT engine with social context reasoning capabilities, we extended the SCoRE. This extension comprises three rules: the goal-oriented, same situation and share resource rule.

6.1 Goal-Oriented Rule

This rule allows our PWT system to identify other people that had fulfilled the user's same personal goal through a PWT system. This information permits PWT applications to consider these other users' web-tasking in order to improve one user's PWT. the inferred pwt context fact is a triple whose predicate corresponds to the object property *pwt:sameGoal* of the PWT Ontology (*pwt* prefix). This inferred information is valid onlyduring the achievement of a particular goal. Figure 9 depicts our conceptual representation for this rule.

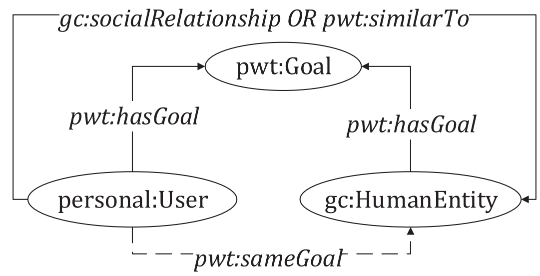

Figure 9: Goal-oriented rule representation

Goal-Oriented Rule:
(?A pwt:hasGoal ?B),
(?C pwt:hasGoal ?B),
((?A gc:socialRelationship ?C) OR (?A pwt:similarTo ?C)),
→ *(?A pwt:sameGoal C?).*

Figure 10 depicts and example for this rule, in which Nina has the same goal as other two users: Lorena and Richard. By previous examples we know that Nina is a colleague of Lorena therefore they share a social connection and Lorena's context becomes available during Nina's PWT. Despite Richard is not socially related to Nina, their similarities about transportation preferences makes Richard's context relevant when Nina executes a web-tasking related to the *book transportation* subgoal. It is worth mentioning that this social connection between Richard and Nina is temporary and last while Nina fulfils her personal goal.

Following our example, the application of this rule to connect Nina and Lorena would be:
(Nina pwt:hasGoal AttendConference),
(Lorena pwt:hasGoal AttendConference),
(Nina personal:colleagueOf Lorena),
→ *(Nina pwt:sameGoal Lorena).*

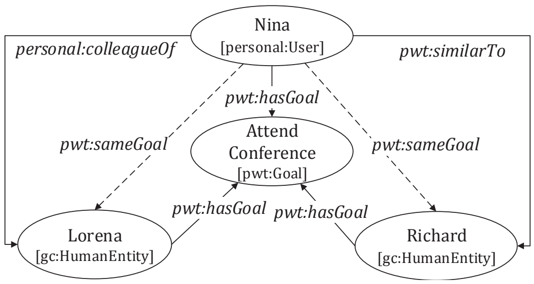

Figure 10: Goal-Oriented rule instance example

6.2 Same Situation Rule

The same situation rule identifies people that participate of the same PWT Situation and are socially relevant for the user's web-tasking. The resulting PWT context fact is a triple whose predicate is identified by the object property *pwt:sameSituation*. Figure 11 depicts our conceptual representation for this rule.

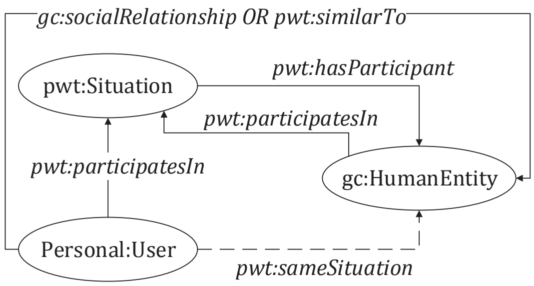

Figure 11: Situation-based rule representation

Same Situation Rule 1:
(?A pwt:participatesIn ?B),
(?B pwt:hasParticipant ?C),
((?A gc:socialRelationship ?C) OR (?A pwt:similarTo ?C)),
→ (?A pwt:sameSituation C?)

Same Situation Rule 2:
(?A pwt:participatesIn ?B),
(?C pwt:participatesIn ?B),
((?A gc:socialRelationship ?C) OR (?A pwt:similarTo ?C)),
→ (?A pwt:sameSituation C?)

Figure 12 illustrates the applications of for this rule, where Nina is in the same situation of two other users: Lorena and Richard. As mentioned above Lorena is in the social sphere of Nina, while Richard only shares similarities.

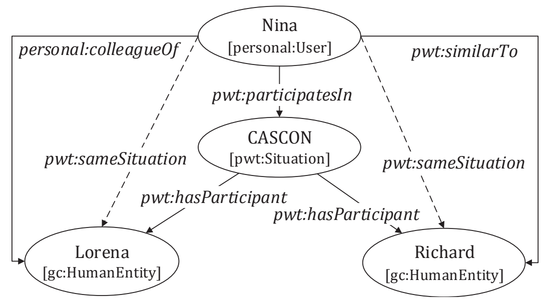

Figure 12: Situation-based rule instance example

(Nina pwt:participatesIn CASCON),
(CASCON pwt:hasParticipant Richard),
(Nina pwt:similarTo Richard),
→ (Nina pwt:sameSituation Richard).

6.3 Shared Resource Rule

In the case of resources, similarities or social context conditions do not apply. That is, users accessing the same resource are by default socially relevant. However, their relevance is only valid during the user's PWT, specifically during the execution of a web-subtask that requires the shared resource. The inferred context fact is a triple whose predicate is object property *pwt:shareResource*. Figure 13 depicts our conceptual representation for this rule.

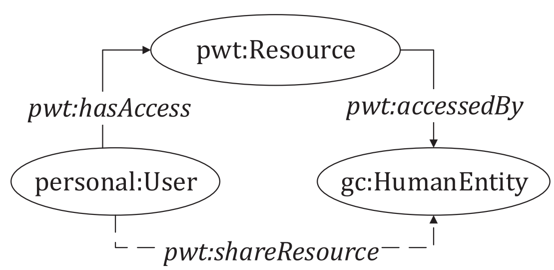

Figure 13: Shared resource rule instance example

Shared Resource Rule:
(?A pwt:hasAccess ?B),
(?B pwt:accessedBy ?C),
→ (?A pwt:shareResource C?).

An example of the application of this rule is depicted in Figure 14 in which a *travel budget file* is accessed by Lorena and Nina. Lorena used this resource during the *"booking transportation"* to define her budget limit when selecting the flight according to a maximum price permitted (cf. Section 3.1 label wst-3). Given that Nina also accesses the same file for the same web-subtask, the PWT system can use Lorena's context to improve Nina's web-tasking. For instance, showing Lorena's choice of flight as the first option for Nina, and removing from the list other flights above the budget limit.

Following our example, the application of the rule would be:
(Nina pwt:hasAccess TravelBudget),
(TravelBudget pwt:accessedBy Lorena),
→ *(Nina pwt:shareResource Lorena).*

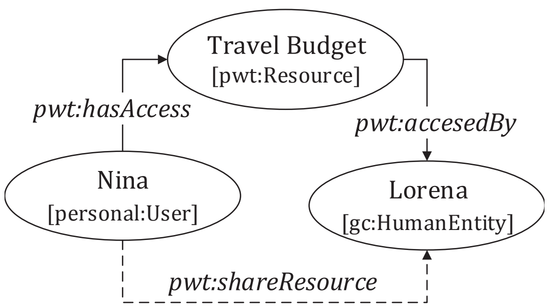

Figure 14: Shared Resource instance example

7 Conclusions

The exploitation of social relationships of the user has been approached by different researchers and applications. From social networks to mobile sensors, many approaches had intended to take advantage of the social interactions of the user to provide diverse functionalities. Personalized web-tasking exploits social context in order to provide personalized features during the web-tasking of the user while fulfilling a personal goal. Consequently, context-awareness has proved to be a fundamental element in the realization of PWT systems.

In this paper we explored some of the relevant approaches for the representation and reasoning of social context, and presented our PWT system and runtime models that support it. Moreover, we showed the social context influence through a scenario based on a group of users fulfilling the personal goal "attending a conference", and described the SMARTERCONTEXT framework proposed by Villegas [24] and its application to our PWT approach.

The contributions of this paper include (1) our definition of *social context* as all information gathered form the social relationships of the user with other users—individually or as a group;

(2) an extension of our PWT ontology to include new classes (i.e., *PWT:Situation* and *PWT:Resource*) as well as object properties such as, *participatesIn*, *associatedWith* and *hasAccess*;

(3) our source-based classification for social context (i.e., explicit, implicit, and sensed);

and (4) our extension to the SCoRE to reason about the relevance of social context in our PWT system.

7.1 Future Work Challenges

To exploit social context effectively in PWT applications we identify the following challenges group them into two concerns: design and technological. We present each challenge with corresponding future work in our research.

Design challenges: These challenges involve design work on our PWT system, runtime models, and the SMARTERCONTEXT framework extensions.

D1. Governance of social context: This implies to incorporate in our PWT system functionalities to manage the life cycle of social context (i.e., the lifespan of the social connection and temporary relevance), and conflict resolution (i.e., the social context gathered from different social connections that conflict with each other).

D2. Measurement: Social context management requires the implementation of strategies to guarantee the relevance of social context in specific situations, and its boundaries (i.e., avoid infinite acquisition and propagation of social context). Also, these metrics that represent lifespan and relevance must be included in our PWT runtime models.

D3. Overhead: This includes required algorithms (or other techniques) to process this social context efficiently when performing web tasks from mobile devices. Also runtime capabilities to decide when the processing must be local or on the web.

Technological challenges: These challenges include those related to the sensors and infrastructures required to deploy PWT applications. We focus on challenges associated with social context.

T1. Instrumentation: Despite the proliferation of context in the *web of things* there is still a limitation to access and exploit information. In fact, social context remains limited to certain sources (i.e., social network platforms and explicit input of the user) and sensor (i.e., mobile devices and software applications that gather specific information). Future instrumentation of the web to exploit context requires standardized mechanisms to represent context entities. We exploit the SMARTERCONTEXT ontology as a first approach.

T2. Social context data exchange: Once there is a standardization to represent context, the following challenge involves the communication protocols to exchange social context. Not to mention, addressing security and privacy concerns. SURPRISE is a policy-based mechanism proposed by Muñoz et al. to secure the access of the personal context of the user in the SMARTERCONTEXT framework [16]. Future work includes to guarantee in SURPRISE the exchange of social context among PWT applications, and the integration of SURPRISE into our PWT System.

Acknowledgment

This work is funded in part by University of Victoria (Canada), IBM Canada Software Laboratory (CAS Research), the National Sciences and Engineering Research Council (NSERC) of Canada under the NSERC Strategic Research Network for Smart Applications on Virtual Infrastructure (SAVI - NETGP 397724-10), and Icesi University (Colombia).

About the Authors

Lorena Castañeda is a PhD student in the Department of Computer Science at University of Victoria, Canada. She is a CAS student at the Center for Advanced Studies at the IBM Toronto Laboratory. She holds a double degree in Engineering: Computer Engineering (2007) and Telecommunications Engineering (2007), and a Master in Information and Communications Management (2012) from Icesi University, Colombia. Her research interests include smart applications, self-adaptive systems, situation- and context-aware systems, user-centric applications, and personalized web-tasking. Email: lcastane@uvic.ca.

Norha M. Villegas is an Assistant Professor at Icesi University, Colombia and Research Associate at University of Victoria, Canada. She received her PhD degree in Computer Science from University of Victoria, Canada. She was a CAS student at the Center for Advanced Studies at the IBM Toronto Laboratory. Her research interests include context-aware self-adaptive software systems, user-centric applications and personalized web-tasking. She received a Diploma Degree in Computer Systems Engineering and a Graduate Degree in Organizational Informatics Management in 2002 and 2004, respectively from Icesi University, Colombia. Email: nvillega@icesi.edu.co.

Hausi A. Müller is a Professor in the Department of Computer Science and Associate Dean Research, Faculty of Engineering at University of Victoria, Canada. He is a Visiting Scientist at the Center for Advanced Studies at the IBM Toronto Laboratory (CAS). Dr. Müller's research interests include software engineering, self-adaptive and self-managing systems, context-aware systems, and service-oriented systems. He serves on the Editorial Board of Software Maintenance and Evolution and Software Process: Improvement and Practice (JSME). He served on the Editorial Board of IEEE Transactions on Software Engineering (TSE). He is Chair of the IEEE Technical Council on Software Engineering (TCSE). Dr. Müller received a Diploma Degree in Electrical Engineering in 1979 from the Swiss Federal Institute of Technology (ETH), Zürich, Switzerland and MSc and PhD degrees in Computer Science in 1984 and 1986 from Rice University in Houston, Texas, USA. Email: hausi@cs.uvic.ca.

References

[1] Silvia Abrahão, Joan Fons, Magalí González, and Oscar Pastor. Conceptual modeling of personalized web applications. In Paul De Bra, Peter Brusilovsky, and Ricardo Conejo, editors, *Adaptive Hypermedia and*

Adaptive Web-Based Systems, volume 2347 of *Lecture Notes in Computer Science*, pages 358–362. Springer, 2002.

[2] Jakob E. Bardram and Thomas Riisgaard Hansen. The aware architecture: supporting context-mediated social awareness in mobile cooperation. In *Proceedings 2004 ACM Conference on Computer Supported Cooperative Work (CSCW 2004)*, pages 192–201. ACM, 2004.

[3] Giulia Biamino. Modeling social contexts for pervasive computing environments. In *Proceedings IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM 2011)*, pages 415–420, 2011.

[4] Lorena Castañeda, Norha M. Villegas, and Hausi A. Müller. Towards personalized web-tasking: Task simplification challenges. In *Proceedings 1st Workshop on Personalized Web-Tasking (PWT 2013) at Ninth IEEE World Congress on Services (SERVICES 2013)*, pages 147–153. IEEE, 2013.

[5] Lorena Castañeda, Norha M. Villegas, and Hausi A. Müller. Self-adaptive applications: On the development of personalized web-tasking systems. In *Proceedings ACM/IEEE 9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2014)*, pages 49–54. ACM, 2014.

[6] Federica Cena, Silvia Likavec, and Francesco Osborne. Propagating user interests in ontology-based user model. In *Proceedings 12th International Conference on Artificial Intelligence Around Man and Beyond*, (AI*IA 2011), pages 299–311. Springer, 2011.

[7] Betty H. Cheng, Rogério Lemos, Holger Giese, Paola Inverardi, Jeff Magee, Jesper Andersson, Basil Becker, Nelly Bencomo, Yuriy Brun, Bojan Cukic, Giovanna Marzo Serugendo, Schahram Dustdar, Anthony Finkelstein, Cristina Gacek, Kurt Geihs, Vincenzo Grassi, Gabor Karsai, Holger M. Kienle, Jeff Kramer, Marin Litoiu, Sam Malek, Raffaela Mirandola, Hausi A. Müller, Sooyong Park, Mary Shaw, Matthias Tichy, Massimo Tivoli, Danny Weyns, and Jon Whittle. Software engineering for self-adaptive systems: A research roadmap. *Software engineering for self-adaptive systems*, Lecture Notes in Computer Science 5525:1–26, 2009.

[8] Martin Halvey, Mark T. Keane, and Barry Smyth. Temporal rules for predicting user navigation in the mobile web. In *Adaptive Hypermedia and Adaptive Web-Based Systems*, volume 4018 of *Lecture Notes in Computer Science*, pages 111–120. Springer, 2006.

[9] I-Ching Hsu. Personalized web feeds based on ontology technologies. *Information Systems Frontiers*, 15(3):465–479, 2013.

[10] Jennifer Horkoff, Eric Yu and Gemma Grau. iStart Guide. Online: http://istar.rwth-aachen.de/tiki-index.php?page= istarQuickGuide. Oct. 2013.

[11] Klemen Kenda, Carolina Fortuna, Alexandra Moraru, Dunja Mladenić, Blaž Fortuna, and Marko Grobelnik. Mashups for the web of things. In Brigitte Endres-Niggemeyer, editor, *Semantic Mashups*, pages 145–169. Springer, 2013.

[12] Rogério Lemos, Holger Giese, Hausi A. Müller, Mary Shaw, Jesper Andersson, Marin Litoiu, Bradley Schmerl, Gabriel Tamura, Norha M. Villegas, Thomas Vogel, Danny Weyns, Luciano Baresi, Basil Becker, Nelly Bencomo, Yuriy Brun, Bojan Cukic, Ron Desmarais, Schahram Dustdar, Gregor Engels, Kurt Geihs, Karl M. Göschka, Alessandra Gorla, Vincenzo Grassi, Paola Inverardi, Gabor Karsai, Jeff Kramer, Antónia Lopes, Jeff Magee, Sam Malek, Serge Mankovskii, Raffaela Mirandola, John Mylopoulos, Oscar Nierstrasz, Mauro Pezzè, Christian Prehofer, Wilhelm Schäfer, Rick Schlichting, Dennis B. Smith, João Pedro Sousa, Ladan Tahvildari, Kenny Wong, and Jochen Wuttke. Software engineering for self-adaptive systems: A second research roadmap. In *Software Engineering for Self-Adaptive Systems II*, volume 7475 of *Lecture Notes in Computer Science*, pages 1–32. Springer, 2013.

[13] Paul Lukowicz, Alex Pentland, and Alois Ferscha. From context awareness to socially aware computing. *IEEE Pervasive Computing*, 11(1):32–41, 2012.

[14] Abderrahmane Maaradji, Hakim Hacid, Ryan Skraba, Adnan Lateef, Johann Daigremont, and Noël Crespi. Social-based web services discovery and composition for step-by-step mashup completion. In *2011 IEEE International Conference on Web Services (ICWS)*, pages 700–701, 2011.

[15] Anmol Madan and Alex Pentland. Vibefones: Socially aware mobile phones. In *Proceedings Tenth IEEE International Symposium on Wearable Computers (ISWC 2006)*, pages 109–112. IEEE, 2006.

[16] Juan C. Muñoz, Gabriel Tamura, Norha M. Villegas, and Hausi A. Müller. 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), pages 131–145. ACM, 2012.

[17] Hausi A. Müller, Holger M. Kienle, and Ulrike Stege. Autonomic computing: Now you see it, now you don't. Design and evolution of autonomic software systems. *Lecture Notes in Computer Science*, 5413:32–54, 2009.

[18] Hausi A. Müller, Joanna Ng, Eleni Stroulia, Kostas Kontogiannis, Norha M. Villegas, and Diana Lau. Personalized Web-Tasking Workshop Organizers' Message. In *Proceedings 1st Workshop on Personalized Web-Tasking (PWT 2013) at Ninth IEEE World Congress on Services (SERVICES 2013)*, page xvii. IEEE, 2013.

[19] Joanna W. Ng and Diana H. Lau. Social ontology and semantic actions: Enabling social networking services for distributed web tasking. In *IEEE Ninth World Congress on Services (SERVICES 2013)*, pages 131–135. IEEE, 2013.

[20] Alex Pentland. Socially aware computation and communication. *IEEE Computer*, 38(3):33–40, 2005.

[21] Charith Perera, Arkady B. Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. Context aware computing for the internet of things: A survey. *IEEE Communications Surveys & Tutorials*, 16(1):414–454, 2014.

[22] José Luís Reis and Joõ Álvaro Carvalho. Aspects that contribute to the success of personalized web applications. In *Advances in Information Systems and Technologies*, Advances in Intelligent Systems and Computing, pages 421–432. Springer, 2013.

[23] Daniel Schuster, Alberto Rosi, Marco Mamei, Thomas Springer, Markus Endler, and Franco Zambonelli. Pervasive social context: Taxonomy and survey. *ACM Transactions on Intelligent Systems and Technoly (TIST)*, 4(3):1–22, 2013.

[24] Norha M. Villegas. Context management and self-adaptivity for situation-aware smart software systems. PhD Thesis, Department of Computer Science, University of Victoria, 2013.

[25] Norha M. Villegas and Hausi A. Müller. Managing dynamic context to optimize smart interactions and services. In *The Smart Internet*, volume 6400 of *Lecture Notes in Computer Science*, pages 289–318. Springer, 2010.