Activity Context Representation: Techniques and Languages
AAAI Technical Report WS-12-05

DCON: Interoperable Context Representation for Pervasive Environments

Simon Scerri[1], **Judie Attard**[1], **Ismael Rivera**[1], **Massimo Valla**[2], **Siegfried Handschuh**[1]
[1]DERI, National University of Ireland, Galway, Ireland
[2]Telecom Italia Labs, Torino, Italia
firstname.lastname@deri.org
massimo.valla@telecomitalia.it

Abstract

Efforts by the pervasive, context-aware system development community have over the years produced a wide variety of context-aware techniques and frameworks. However, a bulk of this technology tends to be strictly tied to a native system, thus largely limiting its external adoption. In addressing this limitation, we introduce an interoperable context representation format, in the form of an ontology, which models core context-aware concepts for re-use within pervasive computing environments. The DCON Context Ontology is proposed as a novel vocabulary for the representation of activity context as experienced by a user, and sensed through one or more of their devices. We demonstrate how, combined with other domain ontologies, DCON provides for richer representations of multi-level context interpretations that are integrated with other known background information about a user.

Introduction

Many initiatives have sought to provide intelligent systems with context-aware features, such as presenting in-context information to the user, automatically executing certain services, and tagging information items with context for later retrieval (Dey 2001). An example of one such initiative is the di.me project[1], which targets the unification of a user's Personal Information Sphere[2] with the aim of providing an intelligent and proactive system, the di.me userware, that assists the user in their day-to-day activities.

Although research in the field has produced a wide variety of context-aware technology, the involved techniques tend to be system-specific, and thus their re-use and/or extension outside the native environment remains limited. In particular, we attribute these limitations to the unavailability of common domain models that represent core context-aware concepts, for their exchange within an infrastructure (Fuchs et al. 2008). In contrast, the di.me approach builds upon an interoperable knowledge representation format for representing, amongst others, context knowledge and context-driven rules. This representation format is provided by a set of ontologies, which consist of "shared conceptualisation" standards based on the Resource Description Framework (RDF)[3]. Earlier research proposes ontologies as the best-suited approach for context modelling in context-aware systems, based on a number of requirements (Strang and Popien 2004).

For its knowledge representation requirements, di.me employs an ontology framework consisting of re-used, extended or newly engineered vocabularies, a majority of which are maintained by the OSCA Foundation (OSCAF)[4]. Based on these ontologies, contextual knowledge that is mined and extracted about the user can be semantically lifted onto a rich, unified and dynamic representation of a users's *Personal Information Model* (PIM), as modelled by the PIMO Ontology (Sauermann, van Elst, and Dengel 2007). Thus, the PIM serves as an integrated personal knowledge base (KB) containing all sorts of personal information about the user (e.g. their devices, accounts, social relationships, resources, activities, etc.). Knowledge stored in the PIM enables advanced features such as distributed personal information management, improved search and retrieval, context-awareness and context-dependant recommendation.

A core vocabulary in the di.me ontology framework is the novel Context Ontology (DCON), which serves as the domain ontology for representing user activity context. Although this ontology has been engineered with particularly the di.me requirements in mind, it remains independent of the project. In this paper, we will demonstrate how the context modelling it provides can by utilised by any system targeting context-aware features in pervasive computing environments. After outlining the high-level modelling decisions we look into for representing activity context in the next section, we present the DCON model in detail and demonstrate a simple example of its use. After a brief comparison to related work, in the final section we then provide some concluding remarks and outline the major future work plans.

[1]http://dime-project.eu/

[2]This term refers to any legacy data, digital footprints and activities that a user possesses, manages and shares across a wide variety of heterogenous personal devices and online accounts

[3]http://www.w3.org/TR/rdf-primer/

[4]http://www.oscaf.org/ – Apart from di.me, OSCAF ontologies have been adopted by various initiatives, including the Social Semantic Desktop (Sintek et al. 2009)

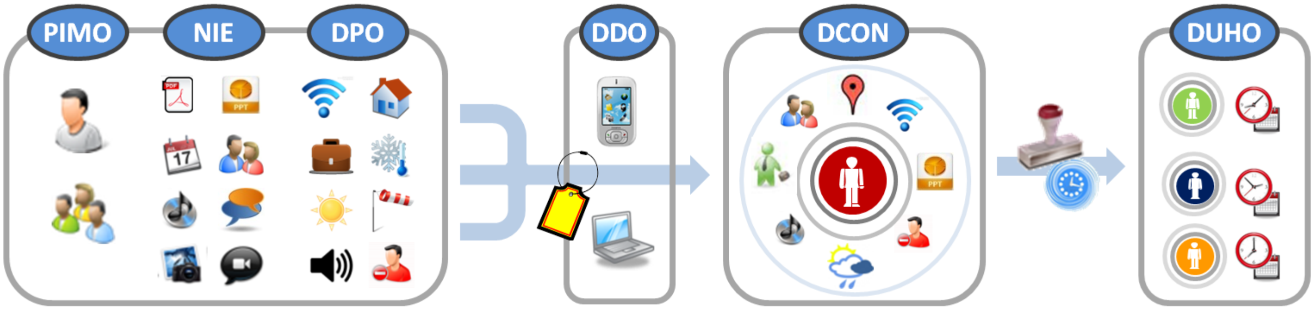

Figure 1: "DCON and friends"—the role of DCON in representing personal context information

Activity Context Definition

Like other pervasive systems, the di.me userware will target different sources for the elicitation of context-related personal information of the user, such as their locations, activities, environments and presence. We differ between two types of sources: sensors, e.g. context information relayed by device-embedded sensors, user attention monitoring, etc.; and social sharing activities, serving as 'virtual sensors' (Cano et al. 2011). Thus, personal devices will serve as a proxy for the user, whereby raw, user activity context that can be sensed through their use is gathered, mashed and interpreted into higher-level abstractions, or what we refer to as *situations*. In fact, a widely-accepted definition of the term 'context' is "any kind of information that can be used to characterise the situation of the user, as an entity (Dey 2001). The advanced context-aware features envisaged for di.me will also be triggered by recurring user situations, and not just low-level activity context triggers.

In line with the above rationale for pervasive systems, we differ between the following three levels of user context information:

i) low-level – 'raw' information that can be directly retrieved from device sensors and system activities (e.g. specific location, time, running applications, speeds and movement, etc.)

ii) mid-level – raw context information that has been interpreted into mid-level, *user presence* concepts (e.g. in a place marked as 'workplace' or 'home', during a particular time period of the day/year, during a 'travelling', or 'working' activity etc.)

iii) high-level situations (e.g. "working in the DERI offices", "working from home", "travelling to the UK").

The DCON ontology models all above context levels, such that higher-level context interpretations are composed of lower-level context information. Context is represented as a named graph (Carroll et al. 2005) containing snapshots of context information from one or more specific points in time. DCON is by no means a standalone vocabulary, in the sense that it is integrated within a comprehensive ontology framework that models all sorts of personal information, ranging from information items belonging to the user, to their online profiles, personal preferences and histories. Although each of these personal information domains (including user context) is represented by a separate ontology, they all refer to concepts and attributes defined within each other. Therefore, to fully understand the knowledge modelling provided by DCON, a brief overview of the other ontologies is required.

Figure 1 outlines the role the DCON plays in representing personal information aspects relating to user activities, context and situations. Low-level context information is extracted directly from personal devices, which are described by the Device Ontology (DDO)[5]. Context data from sensors on each device is mapped directly onto a DCON instance[6]. Apart from datatype values (e.g. temperature, time and date, geographical location, etc.), low-level context data also includes references to existing/known items on the Personal Information Model (e.g. running applications, open files, calendar events scheduled for now, etc.). These items are represented by one of the NIE Information Element domain ontologies[7], each of which models files (NFO), events (NCAL), tasks (NMO), contacts (NCO), multimedia items (NMM), messages (NMO), etc. as stored on a user device. Additionally, the DDO also models context information such as known networks that are in range.

The di.me system will continuously process low-level context information to determine richer semantics. For example, devices that are detected nearby (e.g. via bluetooth) are checked against the list of known devices belonging to the user's contacts. A match would result in a high probability that a contact (or a group of contacts) is nearby, which is then stored in the DCON instance as mid-level context information, by referring to existing representations of persons and groups as modelled in the PIMO ontology (Sauermann, van Elst, and Dengel 2007). Another important new ontology related to context modelling is the Presence Ontology (DPO)[8], which models user presence concepts mentioned above. By the term 'presence', we refer to a loosely-defined physical and/or virtual user circumstance, or state-of-being, whose level of abstraction makes it independent of time, in the sense that it can recur. Thus, various low-level con-

[5]Currently a candidate OSCAF submission

[6]In di.me, we refer to this instance as the 'Live Context'

[7]http://www.semanticdesktop.org/ontologies/nie/

[8]http://www.semanticdesktop.org/ontologies/dpo/

text information can be processed and mapped into specific user presence elements, e.g. discrete context values retrieved from sensors (temperature, brightness, time of day/year) into pre-defined ranges ('Hot', 'Low light', 'Evening', 'Weekend', etc.). The DPO also covers non-quantifiable user presence components that can be interpreted from low-level context[9], such as user activities ('Working', 'Travelling', 'Doing Sports', etc.), availabilities ('Busy', 'Available', etc.), and places ('Home', 'Workplace', 'Restaurant', etc.). Just like items from NIE and PIMO, once any of the existing presence components is determined to be relevant to a user's current context, they are also attached to the DCON instance.

Finally, DCON instances can be stored for history purposes, serving as *snapshots* of previous user contexts. The User History Ontology (DUHO)[10] is here valuable since it allows entire DCON instances to be wrapped as a named graph containing past context information. Without going into system-specific details, in di.me, the storage of these instances can be initiated both by the system (e.g. in privacy-sensitive situations, such as a privacy-preference change by the user) but also directly by the user (e.g. to save what they consider a situation of interest). Context matching techniques to be employed by di.me will continuously check whether the changing context information (low-level, mid-level interpretation) significantly matches any of the stored situations, in order to detect their recurrence. The user's interaction with an interactive intelligent user interface (e.g. confirming a recurring situation, adjusting weights to different attributes for more accurate detection) will eventually enable their 'crystallisation' into generalised, high-level form of user activity contexts.

Context Representation Model

In this section we introduce the vocabulary of the DCON Ontology, focusing on how it is able to represent both low (raw) and mid-level (interpreted) context streaming on from a user's (multiple) devices, as well as higher-level recurrent situations that are learned, or characterised, through a series of past context snapshots.

Representing Low and Mid-level Context

Figure 2 depicts the entire DCON vocabulary; the focus of this section and the main contribution of this paper. At the highest abstraction level (grey box, marked 'A'), an entity's context (instance of *dcon:Context*) contains a number of context *aspects* (*dcon:hasContextAspect*), each of which refers to a particular set of context *elements* (*dcon:hasContextElement*). This categorisation of context information is directly inspired by Schwarz's context modelling (Schwarz 2006), which was modelled user activity context as perceived within a desktop environment. DCON's modelling extends this to cover ubiquitous environments. *dcon:Context* instances are defined as named graphs, and thus the containment relationship between a context and various aspects and their elements can be defined simply by storing that information within a context graph.

dcon:Aspect and *dcon:Element* are abstract classes, and are not meant for direct use. Instead, DCON defines the following seven context aspect subclasses (grey box 'B'):

1) Schedule: representing events and tasks in which the user is known to have been, or soon-to-be, or is participating in (retrievable from task lists and calendar entries)
2) Peers: representing contacts that are known to be nearby (as identified by their devices/network connections)
3) Environment: representing environmental information as provided by device sensors, as well as location-based services such as weather forecast provision.
4) Attention: representing what the user's device activities(running applications, applications in foreground, open documents, etc.)
5) SpaTem: representing spatial and temporal user information, including GPS position, known nearby locations, mid-level context information such as deduced speed and direction, as well as time.
6) Connectivity: representing connections that are in range or connected to from a user device, including phone networks, wireless and wired networks, bluetooth, etc.
7) State: representing physical activity context (e.g. travelling, working, etc.) that can deduced from a user's online activities (e.g. status message, check-ins), availabilities, device modes, etc.

A context aspect is included in a *dcon:Context* graph only if one or more of its allowed context elements are present. Whereas aspects are defined by DCON, elements refer to concepts provided by the other ontologies shown in Figure 1. Thus, specific instances will be attached to each aspect. DCON context elements are shown attached to each of the aspects in the lower section of Figure 2, e.g., *nfo:Application*, *nie:InformationElement*, *ncal:Event*, *pimo:Person*, *dpo:Temperature*, etc. Although the core attributes for these elements are provided by the respective domain ontologies (e.g. application name, file size, event date, person name, temperature range), DCON extends their descriptions with context *attributes* (*dcon:hasContextAttribute*, grey box 'A'), or attributes which are only valid within a particular context/period of time. Examples of these attributes are shown attached to the DCON context elements, e.g. application is in the foreground (*dcon:isForeground*), file is open in read-and-write mode (*dcon:writeable*), network is connected (*dcon:connected*), exact current temperature (*dcon:temperature*). DCON also assigns aspect-independent attributes to each context elements (top-left corner). *dcon:recordedBy* defines which device has recorded the presence of each element, whereas *dcon:recordedAt* stores the occurrence time. The *dcon:validity* property is used to predefine a span of time for each element, during which it is assumed to remain relevant (e.g. active applications context changes faster than the outside temperature, or an event scheduled to last for three hours).

[9]Details of how low-level context can be interpreted into a higher-level of abstraction are currently under review at another conference.

[10]Currently a candidate OSCAF submission

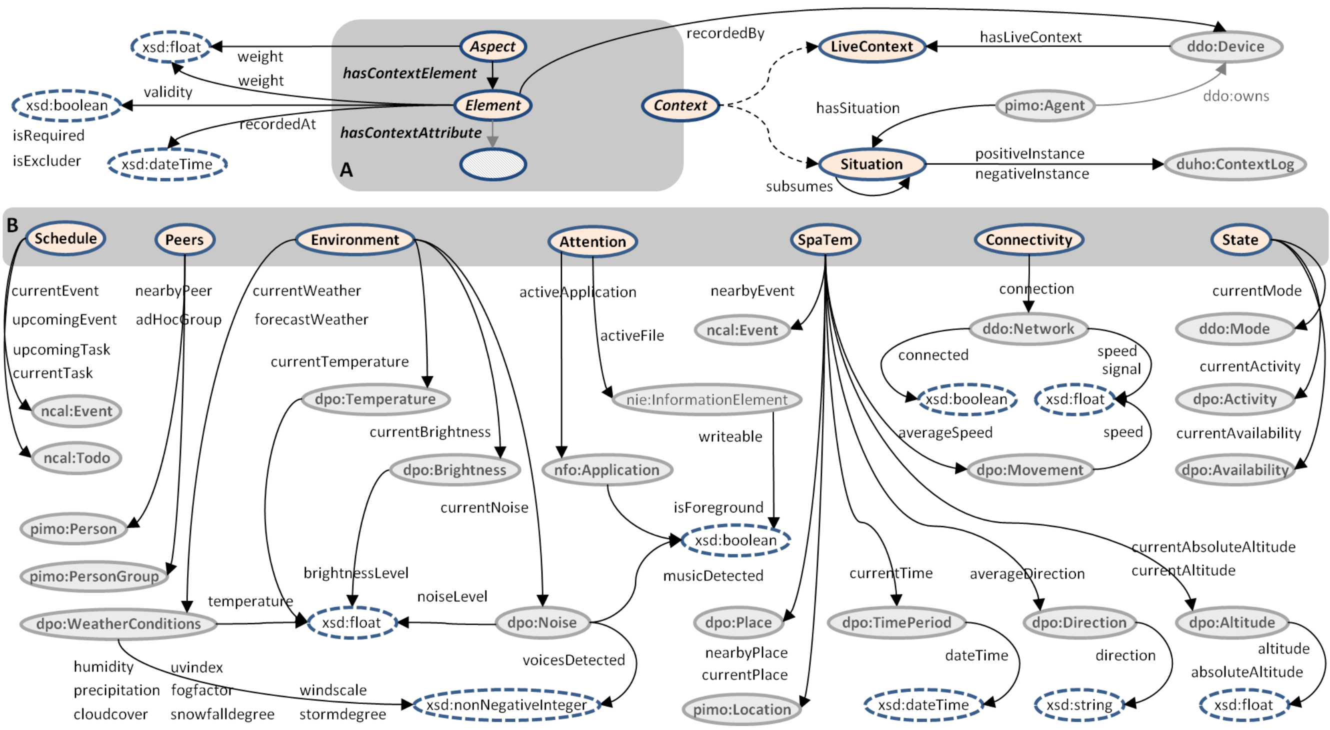

Figure 2: The Context Ontology (DCON)

Whereas some of the elements within a live context graph are directly retrieved from sensors, other contextual elements need to be interpreted by a context-aware system. This distinction corresponds to the first two levels of context information described in the Activity Context Definition section. A specialised di.me component will take care to interpret raw context information (e.g. current temperature, position co-ordinates, position change rate) into more specific, 'fuzzy' mid-level context (e.g. weather category "sunny", place "home", movement "walking speed", direction "NNW"). Although details of this component are outside the scope of this paper, we here emphasize the role of the DCON ontology to represent raw context information, on which the context interpreter will then operate.

Listing 1 shows an example of how an instance of *dcon:Context* can be wrapped within an RDF named graph. The graph shown is marked by the di.me system as an instance of *dcon:LiveContext* in its graph metadata (Listing 2). The context graph itself contains a number of context aspects, each of which refers to a number of context elements. The elements themselves are independent of this context, and are defined in the RDF named graph that contains the PIM (Listing 3). However, within this context graph these items are also instantiated as *dcon:Element* instances, in order to enhance them with context-dependant attributes.

Representing High-level Situations

Another subset of the DCON vocabulary is used to store and characterise high-level situations, which can be best described as a generalised context instance that defines a recurrent yet flexible user activity, based on a number of past live context instances that are semi-automatically marked as positive or negative instances by the user. In di.me, this vocabulary is crucial for the context matching component, which strives to identify recurring user situations.

DCON differentiates between two major subclasses of *dcon:Context* (top-right corner in Figure 2): *dcon:LiveContext*—which as shown by the previous examples mark a context as being 'current'; and *dcon:Situation*—which entail the high-level context information corresponding to the third level described in the Activity Context Definition section. Based on the user's input through the di.me user interface, the system will slowly learn how to characterise a situation, through a number of past live context instances (*duho:ContextLog*) which serve as positive (*dcon:positiveInstance*) and negative (*dcon:negativeInstance*) examples. Thus, a situation may combine context elements from various previous contexts—all of which characterise that situation. This is also reflected in the fact that whereas a lower-level live context instance is attached to a device (ddo:Device), if active, a high-level situation is instead attached to an agent (pimo:Agent) who owns one or more devices.

In order to facilitate the context matching technique, DCON provides vocabulary for the characterisation of situations through context aspect and element weights and markers. The *dcon:weight* property initially allows the user to manually state which aspects best define a manually stored situation. When the context matching determines that a situation is recurring, the user is prompted to confirm or reject

```
@prefix dcon:    <http://www.semanticdesktop.org/
     ontologies/2011/10/05/dcon#> .

#Named graph containing the live context for user Juan
<urn:juan:graph:live-context> {

#Active context aspects and their elements
juan:LiveContextSchedule
    a dcon:Schedule ;
    dcon:currentEvent juan:Event182 .

juan:LiveContextPeers
    a dcon:Peers ;
    dcon:nearbyPeer juan:Person12 , juan:Person23 .

juan:LiveContextAttention
    a dcon:Attention ;
    dcon:activeFile <file:/../plan_bridge_Brown.jpg> ;
    dcon:activeFile <file:/../plan_presentation.ppt> ;
    dcon:activeApplication juan:App14 , juan:App32 .

juan:LiveContextConnectivity
    a dcon:Connectivity ;
    dcon:connection juan:Connect2 , juan:Connect8 .

juan:LiveContextSpaTem
    a dcon:SpaTem ;
    dcon:currentPlace juan:Location14 .

#Other Aspect descriptions
...

#Context attributes attached to PIM elements
<file:/../plan_bridge_Brown.jpg>
    a dcon:Element ;
    dcon:writeable "true" ;
    dcon:recordedBy  juan:Device1 ;
    dcon:recordedAt  "2011-11-25T12:02:34Z" .

juan:App14
    a dcon:Element ;
    dcon:recordedAt  "2011-11-25T12:03:12Z" ;
    dcon:foreground "true" ;
    dcon:recordedBy  juan:Device1 .

juan:Connect2
    a dcon:Element;
    dcon:recordedAt  "2011-11-25T11:08:20Z" ;
    dcon:signal "3.00"^^xsd:float ;
    dcon:recordedBy  juan:Device1 ;
    dcon:connected "true" .

#Other context element descriptions
...}
```

Listing 1: A *dcon:LiveContext* instance

its reactivation. Based on the interaction, aspect and element weights will be adjusted, signifying the context matching components ability to learn about user situations. In addition, DCON also provides two special markers that define which elements are necessary for recognising a particular context/situation (*dcon:isRequired*), and the inverse, i.e. its presence would right away eliminate a particular situation from being matched (*dcon:isExcluder*).

```
#Named graph containing metadata about the live context
     graph above
<urn:juan:graph:live-context/metadata> {

#Live context graph descriptions
<urn:juan:graph:live-context>
    a       dcon:LiveContext ;
    nao:lastModified "2011-10-05T00:00:00Z" ;
    nao:prefLabel "Live Context" ;
    nao:description "A fixed live context instance for
        di.me" . }
```

Listing 2: Metadata for the *dcon:LiveContext* Graph

Listing 5 shows an example of a learned user situation. The situation is defined as an instance of *dcon:Situation* in Listing 4 and labelled “Working in DERI”. It is initially saved by the user from the live context snapshot shown in Listing 1, which is automatically logged to an instance of *duho:ContextLog* (urn:juan:graph:contextlog43) and attached to the situation as a positive instance. Over time, interactions with the di.me user interface enable the system to store additional positive as well as negative instances. The contents of the situation graph in Listing 5 contain a generalisation of context aspects and elements that are common to all these context instances, e.g. people who are usually around when the user is ‘working in DERI’, networks that are usually in range, the location, etc. Aspects have weights, which are adjusted over time to reflect the learned relevance of particular context information in characterising a situation. In this example, the user’s location and the networks in range are both considered to have a high weight (where weight ranges from a low of 0 to a high of 1). In addition the system also knows about some context elements which are required, i.e. without them being active the situation is excluded by the context matching component, or which act as excluders, i.e. if they are active, the situation is excluded. An example of the former is ‘Location14’, whereas an example of the latter is ‘Location13’. These two items are defined in the PIM graph (Listing 3) and interpreted in this context they mean that if the user is in the ‘DERI offices’ then they are bound to be in situation “Working in DERI”, unless they are in a particular area that is known as the ‘DERI cafe’.

Related Work

In an evaluation of context modelling approaches in ubiquitous computing systems (Strang and Popien 2004), ontology-based models were determined to be the most adequate. Context ontologies have been proposed by quite a number of initiatives (Chen et al. 2004; Shih, Narayanan, and Kuhn 2011; Reichle et al. 2008; Schwarz 2006; Gu, Pung, and Zhang 2005; Gómez-Romero, Bobillo, and Delgado 2011). However, our approach is rather unique since it requires coverage for a wide spectrum of heterogenous context information streaming in from distributed devices

```
@prefix juan: <urn:juan:> .
@prefix pimo: <http://www.semanticdesktop.org/ontologies
     /2007/11/01/pimo#> .
@prefix nao:     <http://www.semanticdesktop.org/
     ontologies/2007/08/15/nao#> .
@prefix ddo: <http://www.semanticdesktop.org/ontologies
     /2011/10/05/ddo#> .
@prefix ncal: <http://www.semanticdesktop.org/ontologies
     /2007/04/02/ncal#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

#Juan's Personal Information Model (PIM) graph
<urn:juan:PIM> {
juan:PIM a pimo:PersonalInformationModel .

#Descriptions of PIM items
juan:Event182
    a ncal:Event ;
    nao:prefLabel "di.me Project Meeting";
    #Other core attributes from the NCAL domain ontology
         ..

juan:Person12
    a foaf:Agent , pimo:Person ;
    nao:prefLabel "Anna Alford";
    #Other core attributes from the PIMO domain ontology
         ..

juan:Device1
    a ddo:Device ;
    nao:prefLabel "HTC";
    #Other core attributes from the DDO domain ontology
         ..

juan:Location13
    a pimo:Location ;
    nao:prefLabel "DERI Office";
    #Other core attributes from the PIMO domain ontology
         ..

juan:Location14
    a pimo:Location ;
    nao:prefLabel "DERI Cafe";
    #Other core attributes from the PIMO domain ontology
         ..

#Other PIM item descriptions...}
```

Listing 3: An excerpt from Juan's PIM graph

in an ubiquitous environment. The heterogeneity problem is addressed through an existing set of available ontologies, on which DCON is heavily reliant for the representation of context-dependant attributes of information items. That said, some modelling decisions taken for the engineering of DCON are comparable to some of those approaches. Shih et. al. cite the need to relate context inferred from sensor data with other background knowledge about the user (Shih, Narayanan, and Kuhn 2011). The SOUPA ontology also extends an imported set of ontologies with additional pervasive application domain models (Chen et al. 2004). However, our approach stands out mainly because context information is

```
#Named graph containing metadata about the live context
     graph above
<urn:juan:graph:situation14/metadata> {

#Live context graph descriptions
<urn:juan:graph:situation14>
          a       dcon:Situation ;
          nao:lastModified "2012-05-05T00:00:00Z" ;
          nao:prefLabel "Working in DERI" ;
          dcon:positiveInstance <urn:juan:graph:
               contextlog43> ;
          dcon:positiveInstance <urn:juan:graph:
               contextlog98> ;
          dcon:positiveInstance <urn:juan:graph:
               contextlog147> ;
          dcon:negativeInstance <urn:juan:graph:
               contextlog147> ; }
```

Listing 4: Metadata for the *dcon:Situation* Graph

tightly integrated with other information about the user as known and stored in the PIM instance, as modelled by the other domain ontologies shown in Fig 1 (e.g. PIMO, NIE ontology set, DDO, etc.).

The introduction of different layers of context abstractions in DCON (aspects, elements and attributes) follows similar efforts by (Reichle et al. 2008; Schwarz 2006). Reichle et. al. differed between contextual scopes and elements (Reichle et al. 2008), whereas Schwarz also structured context information into various elements, grouped by aspect. In addition, Schwarz's proposed set of contextual aspects has directly inspired the ones modelled in DCON. Other models which target a comparable set of contextual aspects have been proposed in (Cassens and Kofod-Petersen 2006; Chen et al. 2004). In comparison to these other models, DCON provides the most comprehensive set of context aspects.

At the functional level, DCON's modelling is also tailored towards the representation of three different context information levels: raw, mid-level and high-level context information. Comparable approaches that support interpreted (by context-aware applications) raw context information have been described in (Chen et al. 2004; Gu, Pung, and Zhang 2005). In particular, Chen et. al. proposed the use of Semantic Web technology specifically in order to be able to model a higher-level notion of contextual situations, based on which a user can then define information sharing preferences. Through DCON, we have provided a solid basis for realising this proposal. DCON's flexibility enables the representation of interpreted, high-level context information, or situations, based on which the user can then manage context-driven rules.

We have described how DCON can be used in conjunction with other ontologies, particularly the DPO presence ontology and the DUHO user history ontology, to represent richer representations of user context. In (Gómez-Romero, Bobillo, and Delgado 2011), the authors point out that ontologies may not be appropriate to deal with the representation of vague, imprecise context information, despite its be-

```
@prefix dcon: <http://www.semanticdesktop.org/ontologies
     /2011/10/05/dcon#> .

#Named graph containing situation "Working in DERI"
<urn:juan:graph:situation14> {

#Relevant weighted aspects and elements for this
     situation
juan:Situation14Peers
    a dcon:Peers ;
    dcon:weight "0.6" ;
    dcon:nearbyPeer juan:Person9, juan:Person12, juan:
         Person13, juan:Person18, juan:Person22, juan:
         Person23 .

juan:Situation14Attention
    a dcon:Attention ;
    dcon:weight "0.6" ;
    dcon:activeFile <file:/../plan_bridge_Brown.jpg> ,
    #Additional files that have been active in this
         situation...
    dcon:activeApplication juan:App14 , juan:App32 ,
    #Additional files that have been active in this
         situation...

juan:Situation14Conn
    a dcon:Connectivity ;
    dcon:weight "0.9";
    dcon:connection juan:Connect2 .

juan:Situation14SpaTem
    a dcon:SpaTem ;
    dcon:weight "1" ;
    dcon:currentPlace juan:Location14 , juan:Location3 .

#Other Aspect descriptions...

#Context attributes attached to PIM elements
<file:/../plan_bridge_Brown.jpg>
    a dcon:Element ;
    dcon:recordedBy  juan:Device1 .

juan:Location13
    a dcon:Element ;
    dcon:isExcluder "true" .

juan:Location14
    a dcon:Element ;
    dcon:isRequired "true" .

juan:Connect2
    a dcon:Element;
    dcon:recordedBy  juan:Device1 , juan:Device2 ;
    dcon:isRequired "true" .

#Other context element descriptions... }
```

Listing 5: A *dcon:Situation* instance

ing inherent to several real world domains. They propose the use of ‘fuzzy ontologies’ to obtain approximate matching of a ‘current scenario’ (live context) with stored situations. Similarly, we provide the DPO to support the DCON with the representation of fuzzy context information that may recur and is independent of a specific point in time. Finally, the proposed combination of the DCON and DUHO ontologies enables the modelling and storage of situation and context histories, which is crucial in order to enable context-aware applications to predict future user contexts and activities (Kaenampornpan and O?Neill 2004).

Summary

In this paper we have introduced the DCON ontology, an ontology that represents context information as experienced by an entity, as sensed by one or more devices. By means of an example we demonstrated how the DCON can be used to represent an entity’s activity context, in a standard format that can be processed by any machine that knows how to handle RDF data. The DCON vocabulary fulfils the requirements of the di.me userware—a system that will provide context-aware features to a user in a pervasive setting. However, the vocabulary is independent of di.me, and can be extended as required. Thus, we propose it as a standard model for context knowledge representation in similar systems, targeting the current niche in the availability of context models offering multi-device and platform interoperability. If adopted, the rich, flexible context model provided by DCON, will enable different context-aware systems to interoperate and exchange standard context information and interpretations.

In the future, we will continue extending the DCON ontology with new classes and properties to fulfil any additional context information representation requirements. The development of context-aware features in the di.me userware prototype will serve as a proof of concept for our modelling decisions. Here, we will demonstrate how stored situations, coupled with an intuitive user interface, will enable a user to save context-driven rules for firing situation-dependant actions and recommendations. By reasoning over the stored context RDF data, the intelligent system will then be able to determine re-activation of specific context situations, in order to fire the user-defined rules.

Acknowledgments

This work is supported in part by the European Commission under the Seventh Framework Program FP7/2007-2013 (*digital.me* – ICT-257787) and in part by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (*Líon-2*).

References

Cano, A. E.; Dadzie, A.-S.; Uren, V. S.; and Ciravegna, F. 2011. Sensing presence (presense) ontology: User modelling in the semantic sensor web. In *Proceedings of the ESWC Workshop on User Profile Data on the Social Semantic Web (UWeb 2011)*.

Carroll, J. J.; Bizer, C.; Hayes, P.; and Stickler, P. 2005. Named graphs, provenance and trust. In *WWW ’05: Proceedings of the 14th international conference on World Wide Web*, 613–622. New York, NY, USA: ACM Press.

Cassens, J., and Kofod-Petersen, A. 2006. Using activity theory to model context awareness: A qualitative case study. In Sutcliffe, G., and Goebel, R., eds., *FLAIRS Conference*, 619–624. AAAI Press.

Chen, H.; Perich, F.; Finin, T.; and Joshi, A. 2004. Soupa: Standard ontology for ubiquitous and pervasive applications. In *First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (Mobiquitous 2004).*

Dey, A. K. 2001. Understanding and using context. Technical report, Future Computing Environments Group, Georgia Institute of Technology, Atlanta, GA, USA.

Fuchs, S.; Rass, S.; Lamprecht, B.; and Kyamakya, K. 2008. A model for ontology-based scene description for context-aware driver assistance systems. In *Proceedings of the 1st international conference on Ambient media and systems*, Ambi-Sys '08, 5:1–5:8. ICST, Brussels, Belgium, Belgium: ICST.

Gómez-Romero, J.; Bobillo, F.; and Delgado, M. 2011. Context representation and reasoning with formal ontologies. In *Activity Context Representation*, volume WS-11-04 of *AAAI Workshops*. AAAI.

Gu, T.; Pung, H. K.; and Zhang, D. Q. 2005. A service-oriented middleware for building context-aware services. *Journal of Network and Computer Applications* 28(1):1–18.

Kaenampornpan, M., and O?Neill, E. 2004. Modelling context: an activity theory approach. In *2nd European Symposium on Ambient Intelligence, EUSAI 2004, Eindhoven,*, volume 3295 of *LNCS*, 367–374. Springer.

Reichle, R.; 0001, M. W.; Khan, M. U.; Geihs, K.; Lorenzo, J.; Valla, M.; Fra, C.; Paspallis, N.; and Papadopoulos, G. A. 2008. A comprehensive context modeling framework for pervasive computing systems. In Meier, R., and Terzis, S., eds., *DAIS*, volume 5053 of *Lecture Notes in Computer Science*, 281–295.

Sauermann, L.; van Elst, L.; and Dengel, A. 2007. PIMO - a Framework for Representing Personal Information Models. In Pellegrini, T., and Schaffert, S., eds., *Proceedings of I-Semantics' 07*, pp. 270–277. JUCS.

Schwarz, S. 2006. A context model for personal knowledge management applications. In Berghofer, T. R.; Schulz, S.; and Leake, D. B., eds., *Modeling and Retrieval of Context, Second International Workshop, MRC 2005, Edinburgh, UK, July 31 - August 1, 2005, Revised Selected Papers*, volume 3946 of *Lecture Notes in Computer Science*, 18–33. Springer.

Shih, F.; Narayanan, V.; and Kuhn, L. 2011. Enabling semantic understanding of situations from contextual data in a privacy-sensitive manner. In *Activity Context Representation*, volume WS-11-04 of *AAAI Workshops*. AAAI.

Sintek, M.; Handschuh, S.; Scerri, S.; and van Elst, L. 2009. Technologies for the social semantic desktop. In *Reasoning Web. Semantic Technologies for Information Systems*, Lecture Notes in Computer Science. Springer-Verlag. 222–254.

Strang, T., and Popien, C. L. 2004. A context modeling survey. In *UbiComp 1st International Workshop on Advanced Context Modelling, Reasoning and Management*, 31–41.