Linked Context: A Linked Data Approach to Personalised Service Provisioning

Hong Qing Yu*, Xia Zhao†, Stephan Reiff-Marganiec‡, and John Domingue*
*Knowledge Media Institute, The Open University, Milton Keynes, UK
†Department of Computer Science and Technology, University of Bedfordshire, Luton, UK
‡Department of Computer Science, University of Leicester, Leicester, UK

***Abstract*—Nowadays personalised service provisioning becomes more feasible due to the increasing availability of smart devices, such as smart phones, tablet computers, Personal Digital Assistants and Playstations. These smart devices can dynamically detect the context data and upload them to support other interesting software applications, such as Facebook and Google maps. Context can become richer and more retrievable if links are established for semantically related context data sets. Taking advantage of the recent digital and Web technologies, this paper proposes a novel Linked Context model that applies the Linked Data principles to model and obtain context data from both users and services in one unified framework to support personalised service provisioning at runtime.**

***Keywords*-Service provisioning, Semantic Web, Linked Data, Linked context, personalisation.**

I. Introduction

With the fast development of electronic devices and service technologies, providing suitable services that can consider location, available devices as well as other user-related runtime contextual data becomes highly demanded nowadays.

The definition of context has been discussed in different literatures according to specific domains and applications [1], [2]. By summarising previous work, authors in [3] define a general concept of context and context-awareness as the information that can be used to characterise the situation in which people, places or objects relevant to the interaction between the users and the applications, as well as the users and the applications themselves are considered. Context-awareness becomes a key feature for providing adaptable services, for instance, when the best-suited services are required to be selected according to the relevant context information or when services are required to adapt to context changes during their execution [4].

Meanwhile, the service definition becomes wider as it refers to not only SOAP-based web services [5] but also RESTful services [6], SaaS (Software as a Service) [7] and APIs that can support to all different kinds of devices. Although recent research results on personalised service provisioning are very encouraging, some challenges are still remaining. We list two interesting issues that are addressed in this paper:

1) Context modelling: an ideal model shall contain both user context and service functional and non-functional properties.
2) Context integration: a semantic mediation between heterogeneous context data from different resources shall be provided. It resolves the language ambiguity issues and links existing context to offer richer data set.

Over the last few years, a significant portion of research on the Semantic Web has been devoted to create what is referred to as Linked Data [8]. Linked Data is a way to publish data on the Web in order for machines to understand the explicit meaning of the data. The data are linked to other external data sets, and can in turn be linked from external data sets. Therefore, different data sets can be linked and modelled together. Linked Data are based upon a set of principles, including the usage of HTTP URIs to provide information and allowing access through RDF and SPARQL[1]. Since these principles were outlined, there has been a large uptake, most notably through DBpedia[2] to produce a vast amount of linked datasets on the Web. Last year, Linked Data has been proposed as an approach to publishing and describing services, namely linked services [9]. As a result, the service annotations are part of the Linked Data cloud. The linked services concept also introduces a potential opportunity to indirectly link service context information to user context data without ambiguous issues.

In reaction to the two research challenges, a novel Linked Context model is introduced in this paper to fill the semantic gap between the user context and service properties. Our model adopts Linked Data principles to manage, share and link both user context and service context together for developing a personalised service provisioning system. Since Linked Context model deals with the context data that tend to connect more useful data from the Linked Data cloud, then the scalability of using the Linked Context model is essential to be evaluated. In this paper, we will evaluate the scalability of our Linked Context model based on the implemented prototype and experimental test cases.

This paper extends and refines our previous work [10], [11] in this area. The remainder of the paper is organised as follows: the related work about current state art of context-aware systems and Semantic Web services is discussed in Section II. Our main contribution of Linked Context model

[1] http://www.w3.org/TR/rdf-sparql-query/
[2] http://www.dbpedia.org

is illustrated in Section III. The service recommendation method is explained in IV. A real world scenario and a simulation user interface prototype is implemented for evaluation in Section V. Finally, the conclusion is drawn and future research directions are given in Section VI.

II. Background and Related Work

Although context-aware systems are defined differently depending on different types of applications, they share one important feature that is using context information to improve system performance. In early days, context-aware systems are fundamentally built on RMI and CORBA technologies in a network environment. With large number of services being developed and widely used in the industry, context-aware service provisioning becomes an important issue to enable services to be applied in a suitable situation and used by a suitable person.

Location, which was introduced in the Cooltown project [12] and Jini [13], presents the earlier usage of context information for the service selection. The work can discover and select the service nearest to the user. Nevertheless, the context information is only about the location.

Work in [14] and [15] improves the context information by adding dynamic and static Service attributes. The dynamic service attributes are those characteristics of a service whose values change over time. Otherwise, the attributes are said to be static. Since there are more than one context constraints, they also introduced the weighted vector based aggregation functions for ranking the services and returning the top matching service to the user. However, there are two main drawbacks in these early approaches:

1) no semantic representation for contextual information of services. Consequently, it is impossible to apply more advanced semantic level selection, composition and reasoning.
2) not modelling the user context information. Thus, context-awareness in this sense means service attributes awareness.

Research in [16] tackles service syntactic issue by utilising concepts from the Semantic Web. Services are described by formal Semantic Web Services languages, such as OWL-S [17] and WSMO [18] in the context-aware environment. Later on, lightweight service annotation ontologies (e.g. hRESTs [19], WSMO-lite [20] and Minimal Service Model (MSM) [9]) are proposed.

The major consequence is that the user context and service context start to be separately modelled. For example, authors in [21] make a lot of efforts on defining user context information in details and identifies nine categories but without modelling service context. Separated user and service models have two main limitations:

1) a big knowledge gap between users context and services context. On the one hand, the user context model tends to focusing on the information about users, such as location and devices. On the other hand, the service context model normally only concerns service's information such as input data, output data and non-functional properties. Thus, how users context can relate to services context remains an open question.
2) the service ranking algorithms are only based on services context data without considering the whole context picture of users and services.

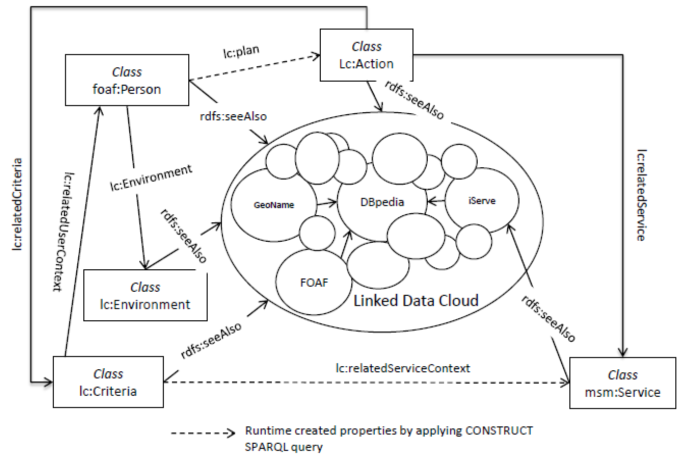


Figure 1. The top layer of Linked Context model

Taking account of the previous related research work and their issues, this paper proposes a novel collaborative Linked Context model, which is based on the Linked Data principles to unify both user and service context in order to better support personalised service provisioning. In this way, all different types of context data are naturally connected without ambiguous confusions. The context is represented as RDF files in a RDF repository and the concepts are identified by HTTP URIs. Context data are linked to each other and referenced by other Linked Data resources in the Linked Open Data (LOD) cloud[3] (e.g France as a location can be referenced as http://dbpedia.org/page/France or be referenced by GeoName[4] as http://sws.geonames.org/3017382/). Each context data can be dereferenced by other linked context data at runtime. The context data (instances) can be reasoned through available Linked Data resources (e.g. using GeoName Linked Data for finding location data and using OpenCyc[5] for finding subtypes). The introduction of the Linked Context brings the advantage of adopting different applications with heterogeneous devices. For example, smart phones may automatically update the user location to Facebook that is one resource the Linked Context model links to. Furthermore, with context data, other useful data may be gathered to support service provisioning. For instance,

[3]http://richard.cyganiak.de/2007/10/lod/
[4]http://www.geonames.org/dataset/
[5]http://sw.opencyc.org/

nearby airports information in the location context will assist the discovery of suitable tickets booking services.

III. Linked Context Modelling and Retrieving

Linked Context model is based on the Linked Data principles and considers three context aspects: the user, the service and the link between them.

In the user context modelling aspect, the most recently promising models are the Unified User Context Model [22] and the three layered ontology context Model [23]. By studying these models, we define three different dimensions of user context.

1) Action context, also named as task context, which states what the user is desiring to do and the final task(s).
2) Profile context that tells the detail information about the user, such as name, age, gender.
3) User environment context that describes the runtime overall situation of the user, such as location, available devices and other available resources.

In the service context modelling aspect, we adopt the MSM[6] as service context model. MSM is introduced by [24] to annotate services by using a lightweight service description ontology and publish services into LOD cloud for semantic service discovery. The two main advantages of using MSM are (1) the ontology is simple to use and focuses on defining a minimum generic model which can be applied to all kinds of services (e.g. SOAP services and RESTful services) and (2) All the instances of the service description can have one or more references (sawsdl:modelreference[7]) to the LOD cloud for explicitly explaining the semantics. In the link aspect, a novel Evaluation criteria context model is introduced.

As result, the Linked Context model is composed by five main ontologies: Action, Friend-of-a-Friend FOAF, User environment, Evaluation criteria and MSM. Figure 1 shows the top level classes of each ontology and the linking relationship or CONSTRUCT queries among them. The CONSTRUCT queries dynamically add links among different parts of the context data at runtime for generating a Linked Context runtime instance. The Linked Context model reuses and extends as much as existing ontologies and previous context modelling research results in order to avoid duplicated concepts. Since the reused ontologies are well defined in their own documentations, the remainder of the paper will only focus on our key contributions in the context models and the integration to supporting personalised service provisioning.

A. Action context model

The most important character of context-awareness is defined as "a property of a system that uses context to provide relevant information and/or service to the user, where relevance depends on the user's task/action" [3]. Action context is always an import aspect in the user context modelling as previous research work suggested (e.g. [22] and [23]). Therefore, we firstly introduce an Action ontology to describe all possible actions that can be performed by registered services. Figure 2 defines that one Action can be a subActionOf another one. Thus, one complex action consists of a group of atomic actions. For example, a travel planning action includes booking transport and booking hotel actions. Each of the action instances are linked to service category class to indicate which kind of service can potentially execute the action (e.g. a hotel booking action is related to a travel service). Meanwhile, each action specifies what facts can influence the service evaluation at runtime (e.g. a hotel booking action is related to the location of the person and the payment method) and reference the action to a semantic vocabulary in the LOD (rdf:seeAlso). When a person plans to perform an action, a reference property (plan) is linked to the matched action instance by using a CONSTRUCT SPARQL query statement shown in Listing 1. The values of userUri parameter and selectedAction parameter are passed to the above query at runtime after a user selecting a desired action. $?u$ is specified as a $foaf$ person and $?a$ is a type of action that is offered by the provisioning platform. The constructed result $lc:plan$ creates a link between user and an action will then be inserted into the original RDF graph that is in the client-side memory.

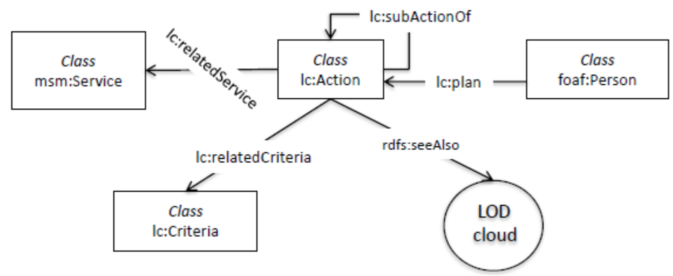


Figure 2. The Action Context Model

Listing 1. SPARQL query

```
CONSTRUCT { ?u lc:plan ?a}
WHERE { ?u rdf:type foaf:Person .
?u foaf:identifier <+userUri+> .
?a rdf:type <+selectedAction+> .
<+selectedAction+> rdfs:subClassOf
lc:Action}
```

B. User-environment context model

The other important model defined is the User environment ontology that links to FOAF[8]. Figure 3 shows the over-

[6]http://iserve.kmi.open.ac.uk/ontology

[7]http://www.w3.org/TR/2007/REC-sawsdl-20070828/

[8]http://xmlns.com/foaf/spec/

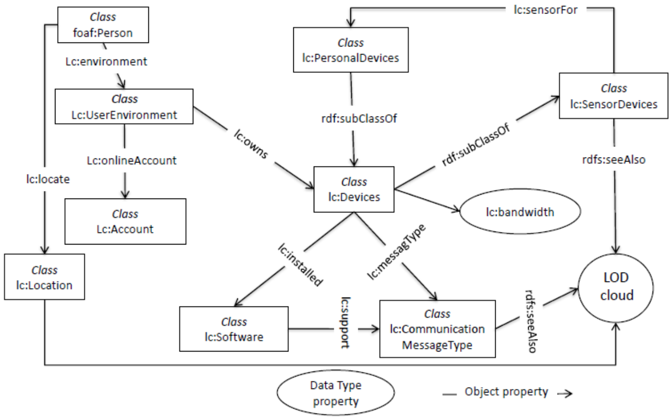


Figure 3. The User Environment Context Model

all model of the User environment ontology that captures both the static and the dynamic information of a user at the time of requesting an action. The User environment model mainly includes user dynamic information about Internet bandwidth, available digital online account, available devices and available software applications with their features. Most of this information can be gathered from devices themselves or reasoned from LOD vocabularies or instances.

C. Criteria context model

The Criteria model (see Figure 4) defines the context-based service ranking constraints and is a core linking point to connect Action, User and Service context. It contains three main classes: Criteria, Criterion and EvaluationType. Each Criteria instance is composed by a set of Criterion instances. For different service category, different Criterion instances are involved. Each Criterion instance specifies what kind of evaluation method (EvaluationType) is suitable for evaluating the related services context data against user context data. Currently, 8 types are defined: (1) numerical, (2) exact, (3) string set, (4) distance, (5) partial matching (6) semantic equal, (7) semantic include and (8) semantic part. If the EvaluationType is specified as the last four types, then the ontology should be referenced. Moreover, each criterion indicates the importance level of the comparison weight inside the criteria.

Each criterion instance dynamically establish links to the user and service context data by applying the predefined CONSTRUCT SPARQL query statements after an action has been initialised. In this way, user and service context data are semantically connected for evaluation the rates of personalisation. The following CONSTRUCT SPARQL statement (Listing 2) is an examples for gathering related user context data to the CoveredLocation criterion that is one factor to affect the service selection.

The CONSTRUCT SPARQL query statement in Listing 2 creates a link between a criterion (denoted as criterionUri input parameter) to a related service property (denoted as ?servceCoveredLocation parameter). The constructed link specifies that the desired service provisioning action related to the criterion that is part of the criteria for evaluating services in a certain service category. More precise, the criterion maps to the service's $coveredlocation$ context property.

D. Linked Context retrieving

We use SPARQL to retrieve context data stored as RDF triples from the RDF repository. We deployed Sesame RDF repository[9] as the backend data storage system that supports SELECT, CONSTRUCT, DESCRIBE and ASK SPARQL functionalities. Firstly, we apply our proposed CONSTRUCT queries to generate a Linked Context instance at runtime. Secondly, we use a SELECT query for getting evaluation required data for service recommendation method. For service described in Section V-A, the SPARQL illustrated in Listing 3 can generate a Linked Context RDF

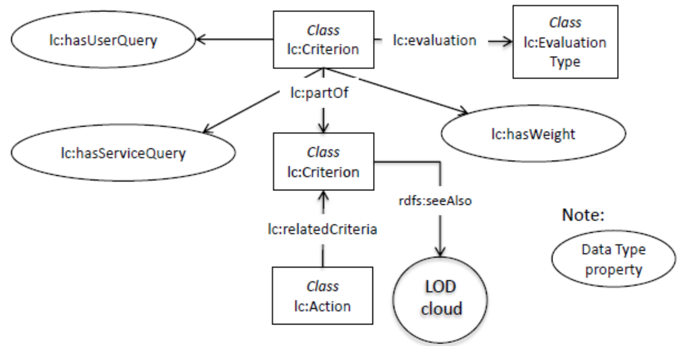


Figure 4. The Criteria Context Model

Listing 2. SPARQL query

```
CONSTRUCT {<+criterionUri+>
lc:relatedServiceContext
?serviceCoveredLocation} Where {
<+criterionUri+> rdf:type lc:Criterion .
?criteria rdf:type lc:Criteria .
<+criterionUri+> lc:partOf ?criteria .
?action rdf:type lc:Action .
?action lc:relatedCriteria ?criteria .
?category rdf:type sf:Category .
?action lc:relatedCateogry category .
?service rdf:type msm:Service .
?service sawsdl:modelreference
?category .
?serviceCoveredLocation rdf:type
wl:NonFunctionalParameter .
?service sawsdl:modelreference
?serviceCoveredLocation .
?serviceCoveredLocation
sawsdl:modelreference ?reference
?reference rdf:type lc:Location}
```

[9]http://www.openrdf.org/

graph for supporting evaluations of the suitable personalised services.

Listing 3. SPARQL query

```
SELECT DISTINCT
?serviceContextValue ?serviceReferences
?userContextValue ?userReferences
?evaluationType ?weight
WHERE { lc:GettingNotification
rdf:type lc:Action .
?criteria rdf:type lc:Criteria .
?criterion lc:partOf ?criteria .
?action lc:relatedCriteria ?criteria .
?criterion lc:relatedUserContext
?userContextValue .
?criterion lc:relatedUserContext
?serviceContextValue .
?userContextValue rdf:seeAlso
?userReferences .
?serviceContextValue sawsdl:modelReference
?serviceReferences .
?criterion lc:evaluation ?evaluationType .
?criterion lc:hasWeight ?weight .
}
```

By performing the SPARQL of Listing 3, all useful data are collected for successfully evaluating services context (in GettingNotification category) against user's runtime context data according to criteria specifications. The Linked Context instance RDF graph for a particular person to a particular action will be not stored persistently in the Sesame RDF repository, but it temperately exists in client side memory space. After the service recommendation, the temperate RDF graph will disappear. Figure 5 shows an RDF graph example of Linked Context instance for Bob to the "Getting Notification" action.

IV. A Type-based LSP Recommendation Method

In our previous work [10], we proposed a service selection (TLE: Type-based LSP Extended) method for introducing a way to use the evaluation type (called Abstract Type) to evaluate different types of criteria. The TLE method addresses the issue of individual criterion evaluation and choosing a suitable mathematical aggregation function. In this paper, we extend our method with four more criteria evaluation functions addressing the semantic perspective in addition to the three original functions to match our defined "EvaluationType". However, to provide a self contained description of the TLE method, we repeat the original three functions in following as well as introducing the extensions.

1) The *numerical* type is used for criteria which take numerical input to the evaluation method such as cost, time and measurement values. The mapped evaluation method is given by Equation 1, where w is the weight of the criterion. When the criterion is of numerical type, the weight can be in the range [-1, 0) or (0, 1]. [-1, 0) means that a smaller numerical value is desired (as e.g. for price properties). v is the value for the service under evaluation, v_{max} is the maximum value of all competitive services for the criterion. v_{min} is the minimum value of all competitive services (if no constraint is specified, it means the lowest value is preferable) or the constraint value of the requirement from the context information.

$$\varepsilon = \begin{cases} \frac{1-(v_{max}-v)}{v_{max}-v_{min}} & \text{iff } w \geq 0, \\ 1 & \text{iff } w \neq 1 \text{and } v_{max} = v_{min} \\ 0 & \text{iff } w = 1 \text{and } v_{max} = v_{min} \\ \frac{v_{max}-v}{v_{max}-v_{min}} & \text{otherwise.} \end{cases} \quad (1)$$

2) The *exact* type is used for criteria which have a certain value evaluated as 1 or 0. The boolean type normally implies an exact match requirement, e.g. the yes/no criteria. It also can be single string and numerical match. The evaluation function is:

$$\varepsilon = \begin{cases} 1 & \text{iff criterion is met,} \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

3) The *string set matching* type is used to define the criteria which are measured by the size of the evaluation objects' satisfaction subset. For example, a person has Visa, Master and Solo cards (this is a value set) and a service supports payment by Visa card only, then the set match level is $\frac{1}{3}$. If the set only contains one value, then the evaluation function becomes essentially identical to the boolean type.

$$\varepsilon = \frac{\varepsilon_1 + \varepsilon_2 + ... + \varepsilon_i + ... + \varepsilon_n}{n} \quad (3)$$

where ε_i is a score for each element of the set.

4) The *partial matching* type is related to a string set matching. However here the criterion is met when at least one part of the criterion value is satisfied. For example, a user has mobile and laptop with Internet access, then any service who can send SMS or email would meet the requirement for contacting the user – the user side might be described by a 'MessageType' criterion (SMS, email).

$$\varepsilon = \begin{cases} 1 & \text{iff at least one value in the criterion is met,} \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

5) The *semantic equal* indicates that the criterion is met semantically according to the ontology mapping schema, although the syntax may be quite different between the criterion value and service context value.

$$\varepsilon = \begin{cases} 1 & \text{iff criterion is semantically met,} \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

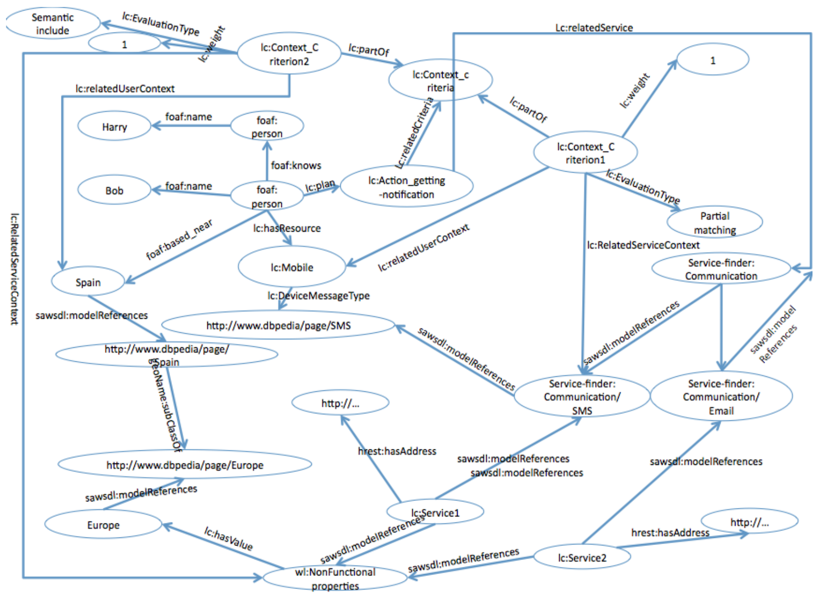


Figure 5. An example of the Linked Context RDF instanace

6) The *semantic include* indicates that if the criterion value is subclass of the service context value based on the particular ontology, then the criterion is met.

$$\varepsilon = \begin{cases} 1 & \text{iff criterion is or subclass of mapped value,} \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

7) The *semantic part* indicates that if the criterion value is superclass of the service context value based on the particular ontology, then the criterion is met.

$$\varepsilon = \begin{cases} 1 & \text{iff criterion is or supclass of mapped value,} \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

The basic idea of TLE method is to use LSP (Logic Scoring Preference) extended aggregation function. The TLE method separates criteria by defining all the hard criteria's weights as 1 or -1 and soft criteria's weights value from [-1, 0) or (0, 1]. This modification can be seen as a further extensions of the weight semantics.

$$E = (\sum_{i=1}^{n} W_i E_i^r)^{\frac{1}{r}} \tag{8}$$

V. Prototype Implementation and Evaluation

A. A motivative Scenario

A context-aware notification service recommendation scenario is described as a user Bob would like to receive company notifications via notification services when he is away travelling on business. The notification services include send email service (ES), send SMS service (SS), send iMessage to iPhone service (i2iS) and send Instant Message service (IMS). Different services have different covered range. For example, ES and IMS are available all over the world, SS is available within Europe and FS is only available in the United Kingdom. The price and response time for each service are also different.

One example runtime situation is that Bob is on the train with only his mobile phone and a software application connecting hotmail instant message installed in the mobile phone. The mobile is currently connecting to Internet via mobile network in Spain. The context-aware service recommendation method should be able to suggest the best suitable sending notification service for Bob based on his dynamic context situation.

B. Prototype implemetation

A prototype is developed for supporting Linked Context management. Figure 6 shows an example of updating a user's "Message Type" context. Users click the "More" button to add more devices and click the "LOD Ref" button to add a classification reference to the added device from the Linked Data cloud. For instance, users may use http://dbpeida.org/page/SMS to reference SMS. As sending notification services use http://dbpeida.org/page/SMS to annotate the service functional classification (see Figure 5), the service recommendation method will benefit to use this semantic matching to do service evaluation. Furthermore, the semantic mediation between different vocabularies can be implemented by comparing the links that different vocabularies point to.

The similar panels are provided for action context, service context and criteria context management.

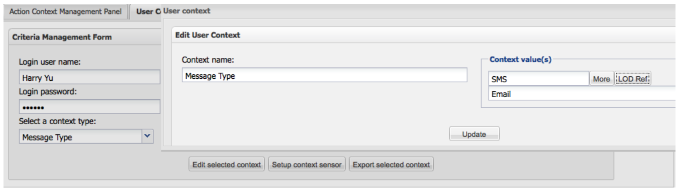
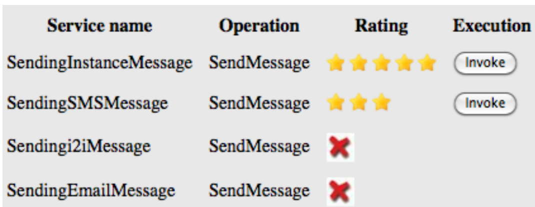


Figure 6. The User context management interface



Service name	Operation	Rating	Execution
SendingInstanceMessage	SendMessage	★★★★★	Invoke
SendingSMSMessage	SendMessage	★★★	Invoke
Sendingi2iMessage	SendMessage	✖	
SendingEmailMessage	SendMessage	✖	

Figure 7. The recommendation result for “Getting Notification” action

C. Prototype evaluation

Applied to our motivated scenario, the ranking results for selecting a suitable service to send a message to Bob are displayed in Figure 7, where the user uses this panel to select an action, get a recommended service list and invoke the selected service.

Two scalability test cases are investigated for evaluating the scalability of the Linked Context model and the recommended approach. The first one is to test the situation when the number of related services is increased from 100 to 1000. The related means that these services are in the exactly same category. When a user selects an action which relates to a category, then all these services are linked to become a graph for service recommendation. To our knowledge, iServe repository stores around 10000 services’ annotations, the related service numbers are normally between 3 to 20. The seekda[10] service repository has around 20000 services. Moreover, 10 is fair number of criteria for normal service recommendation. Therefore, the related service numbers range from 100 to 1000 is reasonable to show the scalability on Web scale.

The evaluation results shown in left-side of Figure 8 indicate that querying Linked Context speed is efficient because time consumption is consistent without affecting by much the growing number of services. Although the timeline grows significantly with growing service numbers by considering whole recommendation approach, it is still scalable as the timeline is close to a linear function.

[10]http://seekda.com/

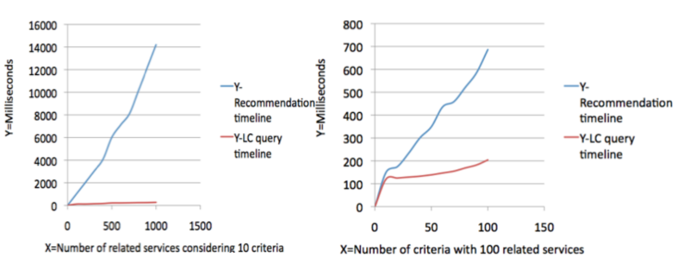


Figure 8. Increasing number of services with 10 criteria

The second test case is to test the situation of increasing number of criteria from 10 to 100 with 100 related services are available. As we discussed that 10 criteria are sufficient for normal scenarios and 100 criteria is actually good number to cover the scalability test for the worst case of the service recommendation scenario. Figure 8 (right-side) illustrates again that the Linked Context is efficient for retrieving required context data at runtime (see the dashed timeline). Meanwhile, the whole service recommendation approach is scalable when criteria is increased because the timeline is close to a linear function.

The evaluation shows that the service selection method is scalable and that good choices are being made. It is however impossible to prove in general that the best services will be selected, as what is best depends on individual users and their needs which can be identified to some extent through the context data but in general there might be some aspect of need that is simple not captured. More crucially, this also means that comparing the method to other service selection approaches that do not sue context is not fruitful, as even less information is available and hence the quality of match of the respective services is not comparable.

VI. Conclusion and Future Work

In this paper, we introduced a Linked Context model for modelling user and service context within one framework. The Linked Context has three major features:

1) The context is modelled using Linked Data principles.
2) The context data are linked together for fulfilling the semantic knowledge gap among different aspects, such as user context and service context.
3) The context data (instances) can be reasoned through available Linked Data services, such as Geoname for finding location data and OpenCyc for finding subtypes and DBpedia identifier of the concept.

The future work will continue investigating more use cases for examining the suitability and usability. The suitability will be evaluated from user’s feedback or tracking whether a user always invokes the top recommended services and whether they satisfy the service execution result. The usability examination will focus on the user interaction interface design. Meanwhile, the plan to investigate more scenarios of different size will allow us to perform better evaluation of the efficiency of the model and approach. We may extend or modify our Linked Context model based on different domain applications. Furthermore, we need to investigate the complexity of the approach from the theoretical point of view to understand the evaluation results in order to improve the application performance. Our model is planned to be integrated into service composition/orchestration processes.

REFERENCES

[1] R. Hull, P. Neaves, and J. Bedford-roberts, "Towards situated computing," in *In Proceedings of The First International Symposium on Wearable Computers*, 1997, pp. 146–153.

[2] D. Franklin and J. Flachsbart, *All gadget and no representation makes Jack a dull environment*. AAAI Press, 1998.

[3] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in *HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*. London, UK: Springer-Verlag, 1999, pp. 304–307.

[4] H.-J. Eikerling, P. Mazzoleni, P. Plaza, D. Yankelevich, and T. Wallet, "Services and mobility: the plastic answer to the beyond 3g challenge," *White Paper, PLASTIC Project*, 2007. [Online]. Available: http://www-c.inria.fr/plastic/dissemination/plastic/dissemination

[5] H. Gunzer and S. Engineer, *Introduction to Web Services*. Borland Developer Network, 2002.

[6] C. Pautasso, O. Zimmermann, and F. Leymann, "Restful web services vs. "big"' web services: making the right architectural decision," in *WWW '08: Proceeding of the 17th international conference on World Wide Web*. New York, NY, USA: ACM, 2008, pp. 805–814.

[7] M. Turner, D. Budgen, and P. Brereton, "Turning software into a service," *Computer*, vol. 36, pp. 38–44, October 2003.

[8] C. Bizerand, T. Heath, and T. Berners-Lee, "Linked data - the story so far," *Special Issue on Linked data, International Journal on Semantic Web and Information Systems*, 2009.

[9] C. Pedrinaci and J. Domingue, "Toward the next wave of services: Linked services for the web of data," *Journal of Universal Computer Science*, vol. 16, no. 13, pp. 1694–1719, 2010, http://www.jucs.org/jucs_16_13/toward_the_next_wave.

[10] H. Q. Yu and S. Reiff-Marganiec, "A method for automated web service selection," in *SERVICES '08: Proceedings of the 2008 IEEE Congress on Services - Part I*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 513–520.

[11] H. Q. Yu and S. Reiff Marganiec, "Automated context-aware service selection for collaborative systems," in *Proceedings of the 21st International Conference on Advanced Information Systems Engineering*, ser. CAiSE '09. Springer-Verlag, 2009, pp. 261–274.

[12] H. Packard, *Cooltown Project*. http://www.cooltown.com/cooltown/, 2004.

[13] N. Klimin, W. Enkelmann, H. Karl, and A. Wolisz, *A Hybrid Approach for Location-based Service Discovery in Vehicular Ad Hoc Networks*. citeseer.ist.psu.edu/klimin04hybrid.html, 2004.

[14] S. Cuddy, M. Katchabaw, and H. Lutfiyya, "Context-aware service selection based on dynamic and static service attributes," in *Proceedings of IEEE International Conference on Wireless And Mobile Computing, Networking And Communications*, vol. 4. IEEE, 2005, pp. 13–20.

[15] C. Lee and S. Helal, "Context attributes: An approach to enable context-awareness for service discovery," in *Proceedings of 2003 Symposium on Applications and the Internet*. IEEE Computer Society, 2003, pp. 22–30.

[16] A.-R. El-Sayed and J. Black, *Semantic-Based Context-Aware Service Discovery in Pervasive-Computing Environments*. in Proc. of IEEE Workshop on Service Integration in Pervasive Environments (SIPE), In conjunction with IEEE International Conference on Pervasive Services (ICPS), pages 556-557, IEEE, 2007.

[17] O.-S. working group, "Owl-s: Semantic markup for web services. member submission," *Member submission, W3C*, 2004.

[18] H. L. Dumitru Roman and U. Keller, "Web service modeling ontology (wsmo)," *WSMO DeliverableD2v1.3*, 2004. [Online]. Available: \urlhttp://www.wsmo.org/2004/d2/v1.0/

[19] J. Kopecký, K. Gomadam, and T. Vitvar, "hrests: An html microformat for describing restful web services," in *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 619–625.

[20] J. Kopecký and T. Vitvar, "Wsmo-lite: Lowering the semantic web services barrier with modular and light-weight annotations," in *Proceedings of the 2008 IEEE International Conference on Semantic Computing*. IEEE Computer Society, 2008, pp. 238–244.

[21] I. Sygkouna, S. Vrontis, M. Chantzara, M. Anagnostou, and E. Sykas, *Context-Aware Services Provisioning on Top of Active Technologies*. Book Series Lecture Notes in Computer Science Publisher Springer Berlin / Heidelberg ISSN 0302-9743, Volume Volume 2881/2003 Book Mobile Agents for Telecommunication Applications, Category Service Management - Service Provisioning pages 67-76 Subject Collection Computer Science, 2003.

[22] F. Carmagnola and F. Cena, "User identification for cross-system personalisation," *Inf. Sci.*, vol. 179, January 2009. [Online]. Available: http://portal.acm.org/citation.cfm?id=1458728.1458873

[23] A. Devaraju and S. Hoh, "Ontology-based context modeling for user-centered context-aware services platform," in *Information Technology, 2008. ITSim 2008. International Symposium on*, vol. 2, aug. 2008, pp. 1 –7.

[24] C. Pedrinaci, D. Lambert, M. Maleshkova, D. Liu, J. Domingue, and R. Krummenacher, "Adaptive service binding with lightweight semantic web services," *In Service Engineering: European Research Results (S. Dustdar and F. Li eds.)*, 2010.