A Rule Based Personalized Location Information System for the Semantic Web

Iosif Viktoratos[1], Athanasios Tsadiras[1], Nick Bassiliades[2],

[1]Department of Economics, [2]Department of Informatics, Aristotle University of Thessaloniki,
GR-54124 Thessaloniki, Greece
{viktorat, tsadiras, nbassili}@auth.gr

Abstract. In this paper, an innovative Personalized Location Information System for the Semantic Web (called SPLIS) is presented. The proposed system adopts schema.org ontology and combines it with rule-based policies, to deliver fully contextualized information to the user of a location-based system. Owners of points of interest can add their own rule-based policies to SPLIS to expose and deploy their marketing strategy on special offers, discounts, etc. These rules are combined at run-time with information about relevant place properties and user (people) profiles. Additionally, owners of points of interest can extend the ontology by adding dynamically specific properties. Rules are encoded in RuleML for interchangeability and to Jess in order to be executed. All data and rules are stored in the form of triples, using Sesame. Rules are evaluated on-the-fly to deliver personalized information according to the rules that fired within the current user-location-time context. In the paper, a demonstration of SPLIS is given using data from Google Places API and Google map for visualization.

Keywords: Rules, Ontologies, Location Based Services, Context.

1 Introduction

Over the last few years Location Based Services (LBS) have become an important part of everyday life, used daily by millions of people for navigation, traffic reporting, sightseeing, etc [1]. Due to the fact that LBS user's environment changes rapidly, it is really important for a successful application to conceive these changes and deliver up-to-date personalized and contextualized information to users [2-6]. If we take as examples a LBS user who is driving and looks for a restaurant close to him, or a LBS user who is looking for a taxi, it becomes apparent that a successful LBS should be capable of offering emerging information retrieval to users, relevant to their needs and preferences. As a result, researches and industries focus on the delivery of such capabilities by developing a) relevant hardware such as GPS, sensors and b) software structures such as semantically annotated metadata for describing the information.

Semantic technologies such as ontologies, gave a huge boost by offering the ability to represent complex user concepts. They enable context perception of higher quality by representing related concepts such as time, location etc. RDFS and OWL are developed to represent the structure of physical entities and the connections between them [7-10]. Furthermore, by providing a general representation standard [8-10], they offer flexibility and interoperability between systems. Connecting and sharing data from various sources (people, places, offers) enables applications that operate on a global data layer and provide more flexible querying as new data appear at runtime.

Ontologies often require reasoning capabilities and they are combined with rule based approaches. Rule-based systems are more autonomous because they are capable of conceiving context changes and respond accordingly without user intervention [11, 12]. Such a capability is important in interactive LBS applications where the user environment is exposed to rapid changes. "Automatic contextual reconfiguration"[2], the ability of a system to keep track of environmental changes and adapt to the current context, is important and leads to high quality services instead of just providing simple information [13-16].

The aim of our work is to combine semantics with LBS to deliver personalized and contextualized information services to users. A system called Semantic Personalized Location Information System (SPLIS)[1] is implemented for this purpose. SPLIS uses an "open" data model by discovering data at run-time. An early version of SPLIS called PLIS has been presented in [17] that implemented a fixed-set of "closed" data model. The use of an "open" data model is achieved by making SPLIS fully compatible with the popular schema.org ontology (also adopted by Google, Bing and Yahoo), incorporating dynamically its RDF Schema version[2]. Because of the variety and the amount of linked data found on the web, a consistently used and widely accepted ontology such as the above is crucial for the success of the system. Additionally in SPLIS, end users and/or place owners are able to enrich the schema dynamically at run time by adding their own properties. Furthermore, SPLIS has a user-friendly form-based web interface for the creation of rules. Another advantage of SPLIS is that its data and knowledge are completely open and interoperable with other systems, being in RDF/S (data and knowledge/schema) and RuleML (knowledge/rules).

After a short review on related work, a detailed presentation of the system is included in the following sections. In sections 3 and 4 the design of our system and the implementation details are described. In section 5 the system operation process is illustrated in detail. In section 6 the functionally of our system is exhibited by the use of a number of examples. Finally, section 7 presents the conclusions of our work and discusses future directions.

2 Related work

Some works from various scientific areas that use semantics are the following:

Point of Interest (POI) recommendation. Serrano et al.[18] proposed a tourist information service which combines RDF data taken from sources such as foaf profile with rules in SWRL format to recommend POIs related with user profile. Noguera et

[1] SPLIS can be accessed at http://tinyurl.com/splis-login

[2] http://schema.org/

al. [19] combined recommendation techniques such as collaborative and knowledge based filtering with smartphones visualization capabilities (e.g. 3D).

Personalized recommendations based on social media data. Savage et al. [20] used foursquare data to make POI recommendations based on a pattern-matching scoring algorithm while Patton et al. [21] implemented a mobile agent for wine and food recommendation. Data from social media such as facebook and twitter are stored in RDF triples format and retrieved for personalized recommendations.

Search optimization. Nowadays, LBS offer high quality mobile search capabilities by personalizing query results or search tag recommendations. Choi et al. [22] proposed a personalized mobile information retrieval system, while Bouidghaghen et al. [23] optimized web search by re-ranking results according to position. Arias et al. [24] proposed a thesaurus-based semantic context-aware auto completion mechanism.

Applications that take advantage of smartphones capabilities. Woensel et al. [25] proposed a person matching application based on data retrieved from foaf profile and identification techniques such as RFID, Bluetooth etc. Furthermore, Keßler et al. [26] combined data collected from sensors with ontologies and rules in SWRL format for utilizing complex context information. Venezia et al. [27], adopted a dynamic rule based approach for personalizing mobile experiences, by offering user the capability to choose from a set of rules (concerning mobile preferences) or add one manually.

Applications to other areas. Except from LBS applications semantic technologies are used in other domains, such as health, e-commerce etc. For example, Bayrhammer et al. [28] used a rule based service bus for controlling data flows in patient centric hospital information systems.

SPLIS is not in contrast with similar works and could easily be combined with most of existing approaches. It offers service providers-place owners the capability to expose their marketing strategy in a direct way. A rule-based approach was followed for SPLIS implementation, based on the advantages discussed in the previous section. Most rule based LBS support a static rule base. Our proposed system differs by enabling a fully dynamic rule base that offers users the option to add rules at run time. By offering users the ability to add rules dynamically SPLIS becomes more intelligent and autonomous. Additionally, a user-friendly interface layer provides proactive personalized information to the regular user.

3 Design and general idea

In everyday life, POIs, as businesses, adopt a rule-based policy in order to deploy their specific marketing strategy (e.g. a coffee shop offers 20% discount to students). On the other hand, LBS users search for such kind of information according to their profile, preferences and needs. For example, someone searches for a cheap coffee shop close to his/her location, or a restaurant having an offer for pizza. SPLIS aim is to provide a direct interaction platform between POI owners and users - potential customers. This can be done by combining business policies with user context and needs in order to deliver up-to-date, personalized information.

Every time a user logs into the system to search for a place, SPLIS gets user's context, evaluates the rules-offers associated with nearby POIs and delivers proactive personalized information to the user, presented on Google Maps[3]. On the other hand, owners of POIs can expose their marketing strategy by inserting their own rule base policy. They are able to assert data concerning the places they own (data based on the ontology referred above, or even new data, by extending the ontology) and rules related to these data (such as reduced prices for unemployed users). SPLIS is currently able to handle rules concerning: a) Every existing property of a POI, b) User's occupation (e.g. a museum offers discount to students), c) User's gender (a pub has special prices for women), d) User's age (free entrance for children under 12 years), e) User's location (a coffee shop decreases coffee price for users who are less than 400 meters away), f) Time (a restaurant is closed on Sundays or a coffee shop has special prices between 17:00-19:00).

4 Implementation and technical details

SPLIS implementation is based on Java Server Pages (JSP) technology, a web development technology based on Java programming language [29]. It was chosen due to the fact that the vast majority of available rule engines are java-based. Furthermore, JSP could be easily integrated with technologies used for RDF, RDFS and OWL storage and querying such as Sesame, Jena, Protégé etc [30, 31]. Another core component of the proposed system is Sesame, which is a very popular choice for RDF data manipulation, such as storing, querying and inferring large sets of RDF triples [30]. Additionally, it can easily be embedded into java-based applications.

An important part of our system is the rule execution process. A rule representation language is necessary to represent human understandable policies. In our case, RuleML and being more specific, Reaction RuleML (a clone of RuleML) was adopted. RuleML (launched August 2000) is a family of sublanguages which are used to publish rules on the web [32, 33] and their main objective is to provide a general rule markup approach for semantic web applications [34]. It is a powerful markup language (XML with a predefined schema) which supports various types of rules such as deductive, reactive and normative. It also addresses the issues of interoperability and flexibility among different systems on the web, by allowing rules to be encoded in a standard way [32]. Moreover, there are attempts to translate natural language to an XML representation [35], therefore the development of a user-friendly interface is foreseeable. Before adopting RuleML we have considered alternative web rule languages, such as SWRL. However, SWRL employs open world reasoning without default negation, while our approach needs close world reasoning. RIF-PRD is also a candidate, currently not supported by tools as much as RuleML [36]. A rule representation language has to be translated into a machine readable language. Rule engines such as Jess, Drools, Prova [37] are used for this purpose. Jess was chosen to implement the core of SPLIS, because of the fact that it is a lightweight rule engine that

[3] http://maps.google.com

connects well with web technologies, which were needed for SPLIS system implementation [38]. Rules in RuleML format are transformed to Jess rules by using XSLT. Additionally, apart from the PC browser-based version of SPLIS, a lightweight mobile version for Android devices has been implemented. The mobile application supports all functionalities concerning regular user operations (e.g. search for POIs, rating, etc.) which are described in the following section.

5 SPLIS operation process

The SPLIS operation process includes the following layers (**Fig. 1**):

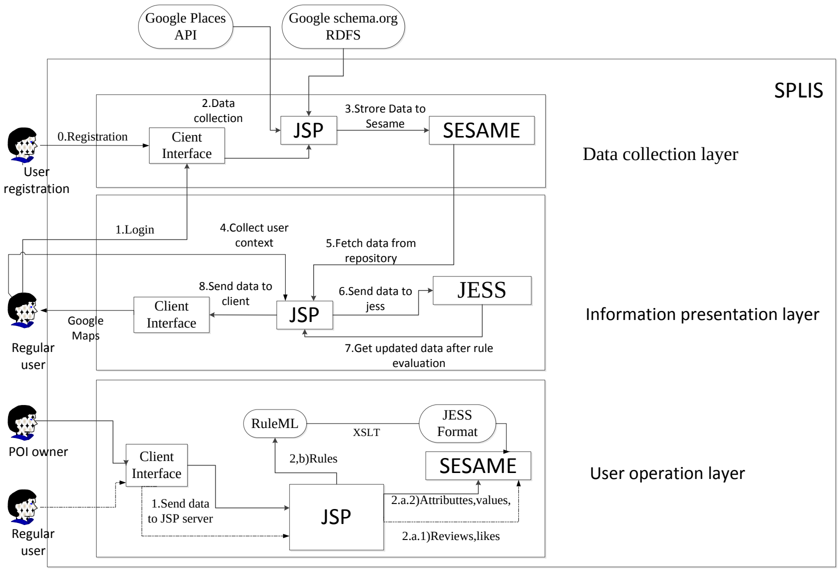


Fig. 1. SPLIS operation process

Layer 1: Data collection. This layer is responsible for gathering and adding new data into the system repository. Concerning user-related data, users are capable of registering to the system by completing a registration form. Form fields are inferred dynamically from the ontology related classes and after submission property values are stored as triples in the repository. After completing the registration process, the user is able to log into the system and (if authentication is successful) a process concerning POI data collection begins. Two steps are included:

1. *Ontology loading.* The ontology is loaded dynamically into Sesame so that the system is always up-to-date with the latest official updates of the RDF Schema.
2. *Data update.* After the completion of ontology loading step, the system obtains user's position (via GPS in mobile or via IP address in desktop) and retrieves the nearest POIs (for reduced computational cost) from external sources such as Google Places API. We manually designed a mapping between the schema.org

RDFS ontology classes/properties and Google Places API categories/attributes based on terms similarities. Specifically, every category is assigned to a related schema.org class (e.g. API "lodging" is matched with schema.org "LodgingBusiness" etc.) and every attribute which is being parsed from the API is mapped to the related schema.org property (e.g. "name" with http://schema.org/name). Data from Google Places API are stored into the RDF repository. If a POI was already stored into the system by a previous use, SPLIS updates its related data with the latest information. For optimization, SPLIS compares existing data with the new one and updates them only if they are different.

Layer 2: Information presentation layer. In this layer, the information is presented to the end user according to his profile and the rules that have been fired. The following steps are included:

1. *Data retrieval.* After completing the data collection process, the JSP gets the current user profile data from the repository, and along with contextual property values (location, time etc.), builds his/her context. Afterwards, for every place, existing data such as property values and rules (if any) are being fetched by the JSP.
2. *Rule evaluation.* Data mentioned above and user context property values are asserted to the Jess rule engine, which evaluates rules and updates POIs property values according to the rules fired. Jess checks the conditions of a rule and concludes whether or not to modify the values of the properties involved in the RHS (THEN) part. The new derived information is fetched by JSP for presentation.
3. *Presentation of personalized information.* Finally, data transfer to client is performed for visualization through Google maps. A user-friendly interface has been implemented so that the user is able to comprehend easily the general idea of SPLIS and find quickly an associated offer. First of all, different markers are used for better illustration. Except from the standard red marker for POI representation, a) a yellow marker indicates that the place contains a rule base (provided by the owner) but no rule was fired for the user, b) a green marker indicates that the place contains a rule base and at least one rule was fired, and c) a crown over the marker indicates that the current user is the POI owner. Moreover, by clicking in a marker, the user can obtain additional information explaining either which rules were fired and why or (if no rule was fired) which rules exist for the place (if any). Additionally, the user can directly execute a number of operations (described in layer 3 below) by clicking the corresponding buttons on the available menu.

Layer 3: User operation layer. In this layer, all supported operations of SPLIS are provided to the user. These operations are related to two different types of users, a) the regular user and b) the user who owns a POI and is able to modify it (POI owner).

1. *Regular user operations*. To begin with, a regular user, apart from seeking for information, can also directly interact with the available POIs. He/she can contribute by writing a review, rating them, or adding "likes". In other words, users not only provide useful information to other people, but they also create information that

can be used during rule creation (e.g. place average rating, number of likes). In addition, users are able to report abuse (e.g. for fake prices) for security and reliability. Moreover, a user is able to insert his/her own POIs (if any).

2. *POI owner operations*. If a user is POI owner, then he/she can update POI's profile by adding new data to its properties through a form-based interface. Properties are dynamically generated from the RDF Schema depending on the place type. Except from asserting data such as property values, the user can also extend the RDFS ontology by adding new properties through a simple form interface. Beside the name, the user can choose through a drop down menu if the added property is related only to a specific place (it is stored as a triple) or if it should be attached to a POI category and its subcategories (it is stored as a property of the related place type-class). An owner can also choose the type of the property among text, number and date.

One of the core functionalities of SPLIS is rule creation. A user-friendly form-based rule editor has been implemented so that users can easily create rules through completing web forms. More specifically, users are able to create rules which pose conditions that relate any property of the specific POI with any property regarding user context. Data collected from the forms are being transformed to RuleML, so that policies and SPLIS knowledge can be re-used by other systems. Afterwards RuleML is transformed to Jess in order to become machine executable, as discussed above.

For example the rule "If day is Tuesday and person jobTitle is student then entrance fee is 5 euro" is represented in Jess as:

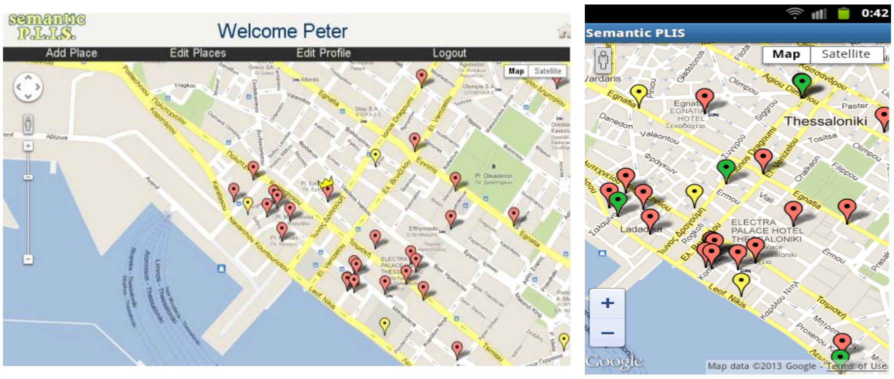
```
(defrule for_students
   (declare (salience 10))
   (person ( day Tuesday)( jobTitle student))
=>
   (modify ?fact (entrance_fee 5))
   (store EXPLANATION "entrance fee for Students on Tuesday's is 5 euro"))
```

In detail, the JESS salience operator is used for resolving rule conflict issues, "modify" changes the values of the related properties and "EXPLANATION" is a variable for storing the explanation and presenting it afterwards to the end user.

Place owners are able to repeat this process and add multiple rules. In case of rule conflict, a LIFO system is used. Furthermore, the same form-based interface is provided to owners for updating directly the existing rules. Alternatively, using RuleML, a POI owner can develop a rule base outside the system and upload it to SPLIS via a URL. Concerning security issues, it's worth mentioning that the editing of the POIs rule base is authorized only to the owner. Additionally, files containing the rules are securely kept to the server.

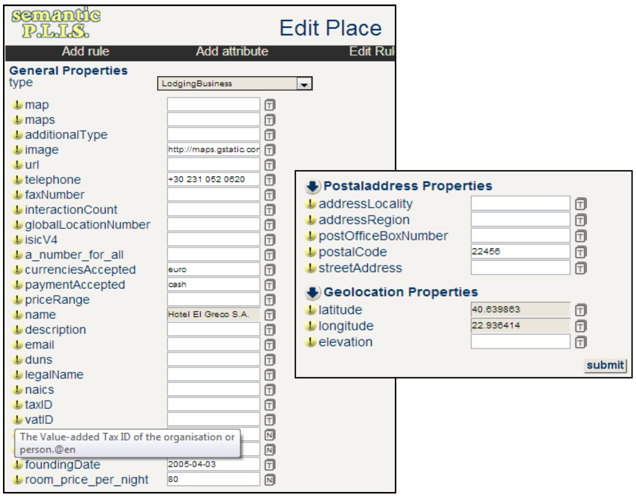
6 Demonstration of SPLIS

To demonstrate the functionalities of SPLIS, let's imagine user "Peter" who is the owner of POI "Hotel el Greco". After entering SPLIS, "Peter" receives the information presented in Figure 2a. He can edit all properties of "Hotel el Greco" as it can be seen in Figure 3. For the correct insertion of data, property types are pulled directly from the RDF Schema and displayed next to each field of the form. Additionally, the POI owner can click on the info icons to receive a textual comment (fetched by schema.org) for the specific property. Notice that data collected from the Google Places API are read-only. By filling the corresponding fields in a web-form, an owner can also add new properties, as discussed above. Let's assume that "Peter" added a new property called "room price per night", which is a number, has domain only "Hotel el Greco" and default value 80.



a) PC browser-based version for "Peter" b) mobile version for "John"

Fig. 2. SPLIS personalized information



General Properties

type	LodgingBusiness	
map		T
maps		T
additionalType		T
image	http://maps.gstatic.cor	T
url		T
telephone	+30 231 052 0620	T
faxNumber		T
interactionCount		T
globalLocationNumber		T
isicV4		T
a_number_for_all		T
currenciesAccepted	euro	T
paymentAccepted	cash	T
priceRange		T
name	Hotel El Greco S.A.	T
description		T
email		T
duns		T
legalName		T
naics		T
taxID		T
vatID		T
		N
		N
foundingDate	2005-04-03	T
room_price_per_night	80	N

The Value-added Tax ID of the organisation or person.@en

Postaladdress Properties

addressLocality		T
addressRegion		T
postOfficeBoxNumber		T
postalCode	22456	T
streetAddress		T

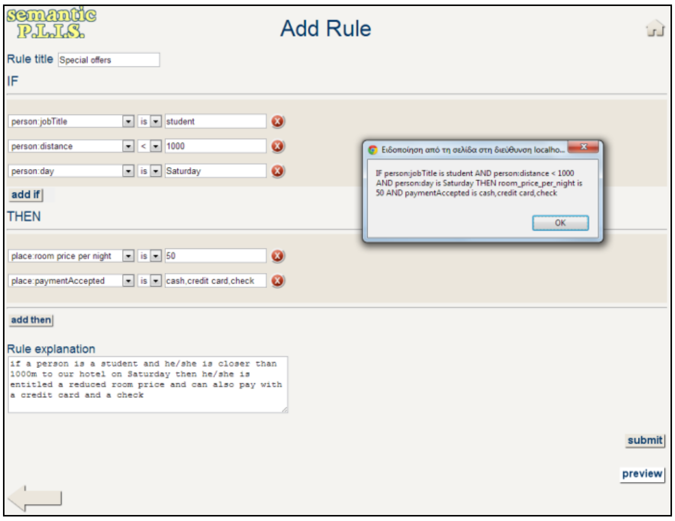
Geolocation Properties

latitude	40.639863	T
longitude	22.936414	T
elevation		T

submit

Fig. 3. Editing form of POI properties in SPLIS

After completing the relevant data entry, "Peter" is able to use the rule editor and add rules. A demonstration of rule creation in SPLIS is given in Figure 4, where the owner is ready to assert the rule "*If a person is a student and day is Saturday and* *distance is less than 1000m then room price per night is 50€ and payment accepted is cash, credit card and check*". The rule editor is horizontally split into two parts, the "IF" part and the "THEN" part. "IF" part elements consist of a property field (place- or user-related), an operator ("is" for text and "<",">" for numbers and dates) and a value field. "THEN" part elements consist of a place property, an assignment operator "is" (or a predefined function "discount" for numbers) and the value field. Additionally, in order SPLIS to provide comprehensible information to the end-user, the web interface provides fields for entering a rule title and a textual explanation of the rule. Also a preview is provided so as the user to check the rule before submitting it.



Rule title Special offers

IF

person:jobTitle is student

person:distance < 1000

person:day is Saturday

THEN

place:room price per night is 50

place:paymentAccepted is cash,credit card,check

Rule explanation

if a person is a student and he/she is closer than 1000m to our hotel on Saturday then he/she is entitled a reduced room price and can also pay with a credit card and a check

Ειδοποίηση από τη σελίδα στη διεύθυνση localho...

IF person:jobTitle is student AND person:distance < 1000 AND person:day is Saturday THEN room_price_per_night is 50 AND paymentAccepted is cash,credit card,check

Fig. 4. Web based rule editor of SPLIS

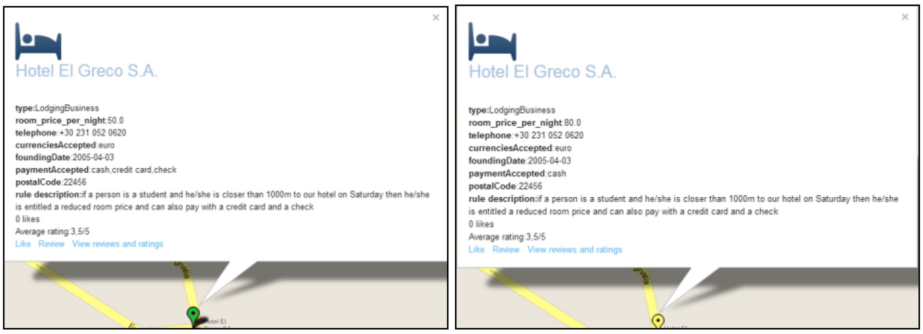
Let now connect POI "Hotel el Greco" with two random regular user profiles who logged in. The following two different profiles snapshots are considered.

1. User A ("John") is a 20-year old student, his current environmental snapshot is taken on Saturday, at 15:14 and we assume his current location (a random location A) is closer than 1000 meters to "Hotel el Greco".
2. User B ("Bob") is 36, unemployed, and has logged in the system on Sunday at 21:15 at a location B (hypothetically closer than 1000m to "Hotel el Greco").

As soon as "John" logs into the system, SPLIS gets his profile, evaluates rules and displays proactive personalized information according to which rules were fired (Figure 2b illustrates the starting screen). By clicking into green markers (POIs where at least one rule is fired) or yellow markers (POIs that contain at least one rule, with no rule to have been fired) he can easily see which rules were fired and why.

For example, if "John" clicks on the green marker of "Hotel el Greco", he receives all the information that is presented in Figure 5a. Because of the fact that "John" is a student, it is Saturday, and John's current location is closer than 1000m from the hotel, the corresponding rule is fired. As a result, a) the room price per night value for “John” has been changed to 50 euro and b) payment accepted options are not only cash (as it is for the default customer) but he can also pay by credit card or check. Additionally “John” receives information about the hotel such as the currency that it accepts, the date that it was founded, average rating etc. “John” can also add a “like” to the POI, add a review and a rate to the POI or view reviews and rates for the hotel that have been submitted by other SPLIS users.

Similarly, when User B - “Bob” logs into the system, no rule is fired for him for this place, because he is neither a student, nor it is Saturday. As a result, the hotel is displayed with a yellow marker and if it is clicked the information presented in Figure 5b appears. Bob is entitled for the standard room price and payment options. Additionally, the rule explanation field displays all the rules concerning “Hotel el Greco” in order Bob to understand a) what kind of rules are contained and b) the reason(s) why they were not fired for him.



Hotel El Greco S.A.

type:LodgingBusiness
room_price_per_night:50.0
telephone:+30 231 052 0620
currenciesAccepted:euro
foundingDate:2005-04-03
paymentAccepted:cash,credit card,check
postalCode:22456
rule description:if a person is a student and he/she is closer than 1000m to our hotel on Saturday then he/she is entitled a reduced room price and can also pay with a credit card and a check
0 likes
Average rating:3,5/5
Like Review View reviews and ratings

Hotel El Greco S.A.

type:LodgingBusiness
room_price_per_night:80.0
telephone:+30 231 052 0620
currenciesAccepted:euro
foundingDate:2005-04-03
paymentAccepted:cash
postalCode:22456
rule description:if a person is a student and he/she is closer than 1000m to our hotel on Saturday then he/she is entitled a reduced room price and can also pay with a credit card and a check
0 likes
Average rating:3,5/5
Like Review View reviews and ratings

Fig. 5. Personalized information for a) “John” and b) “Bob”, regarding Hotel el Greco

7 Conclusions and Future Work

SPLIS was designed and implemented, offering a boost to the quality of delivered information, by giving users the opportunity to add rules dynamically to a location-based information system. On one side, owners add data and rules-offers, and on the other regular users enjoy proactive contextualized information. The capability of adding rules on the fly can not only lead to powerful, autonomous and intelligent services, but also to the evolution of these services. All the above has been achieved by SPLIS that is an “open” system that a) uses the schema.org ontologies, b) uses web based rule editor to create rules that initially are in RuleML format and then they are transformed to Jess rules in order to be able to be executed by the Jess inference engine, c) stores data and rules in the form of triples using the Sesame repository, d) retrieves data using Google Places API and e) display information using Google map. Experimental testing, confirmed SPLIS evolution without developers intervention, as more and more users add rules to the system. The more rules were added to the system, the more interesting and intelligent it becomes.

SPLIS implementation can evolve in various ways. One such way is SPLIS to crawl into POI websites in order to retrieve related RDF data so as POI owners will be able to add data without even logging into SPLIS. Furthermore, not only POI owners but also regular users should be able to have rules-contextualized preferences, as in [39]. Last but not least, in order to resolve possible rule abuse issues, system will check at compile-time the uniqueness and the firing scope of the rule.

8 References

1. S. Steiniger, N. Moritz, E. Alistair. Foundation of Location Based Services. Lecture Notes on LBS (2006)
2. K. Dey, G.D Abowd. Towards a Better Understanding of Context and Context-Awareness. Workshop on the What, Who, Where, When, and How of Context-Awareness (2000)
3. Y.Liu, E.Wilde: Personalized Location-Based Services. Proc. of the iConference (2011)
4. Bouidghaghen O., Tamine L., Boughanem M. Context-Aware User’s Interests for Personalizing Mobile Search.12th IEEE Int. Conference on Mobile Data Management (2011)
5. M. Hosseini-Pozveh, M. Nematbakhsh, N. Movahhedinia. A multidimensional approach for context-aware recommendation in mobile commerce. (IJCSIS) International Journal of Computer Science and Information Security, Vol. 3, No. 1 (2009)
6. K. Kwon, C. Kim. How to design personalization in context of customer retention: Who personalizes what and to what extend? Electronic Commerce Research and Applications, no. 11, pp. 101-116, (2012)
7. H. Hwang , S. Shin, K. Kim, S. Lee, Chang-soo Kim. Context-aware System Architecture using Personal Information based on Ontology. 5th ACIS International Conference on Software Engineering Research, Management & Applications (2007)
8. C. Bizer, Tom Heath, and Tim Berners-Lee. Linked data - the story so far. International Journal on Semantic Web and Information Systems, 5(3):1–22, (2009)
9. N.Tryfona, D. Pfoser: Data Semantics in Location-based Services. Journal on Data Semantics III, vol. 3534, pp. 168–195 (2005)
10. S. Ilarri, A. lllarramendi, E. Mena, A. Sheth. Semantics in Location-Based Services. IEEE Internet Computing (Vol. 15, No. 6) pp. 10-14) (2011)
11. S. Wu, C. Chang, S. Ho, and H. Chao. Rule based intelligent adaptation in mobile information systems. In Expert Systems with Applications, vol. 34, pp. 1078–1092. (2008)
12. Lassila, O. Applying Semantic Web in Mobile and Ubiquitous Computing: Will Policy-Awareness Help? Semantic Web and Policy Workshop (2005)
13. Emmanouilidis, C., Koutsiamanis R., Tasidou, A.: Mobile Guides: Taxonomy of Architectures, Context Awareness, Technologies and Applications. Journal of Network and Computer Applications, p.p.103-125,(2012)
14. R.Etter, P. Dockhorn Costa,T. Broens. A rule-based approach towards context-aware user notification services.ACS/IEEE Int. Conference on Pervasive Services(PERSER) (2006)
15. Goix L. M. Valla, L. Cerami, P. Felcarin. Situation Inference for Mobile Users: A Rule Based Approach, Int. Conference on Mobile Data Management, 2007. pp. 299-303 (2007)
16. S. Wu, C. Chang, S. Ho, and H. Chao. Rule based intelligent adaptation in mobile information systems. In Expert Systems with Applications, vol. 34, pages 1078–1092. (2008)
17. I. Viktoratos, A. Tsadiras, N. Bassiliades, Personalizing Location Information through Rule-Based Policies. 6th International Symposium on Rules: Research Based and Industry Focused (RuleML 2012), Springer, LNCS 7438, pp. 215-223,France(2012).

18. D.Serrano, R. Hervás, J. Bravo. Telemaco: Context-aware System for Tourism Guiding based on Web 3.0 Technology. International Workshop On Contextual Computing and Ambient Intelligence in Tourism (2011)
19. Jose M. Noguera, Manuel J. Barranco, Rafael J. Segura, Luis Martinez. A Mobile 3D-GIS Hybrid Recommender System for Tourism. *TR-1* (2012)
20. N.S. Savage, M. Baranski, N.E. Chavez, T. Höllerer: I'm feeling LoCo: A Location Based Context Aware Recommendation System.8th Int. Symposium on LBS, Vienna, (2011)
21. Patton, E.W. and McGuinness, D.L. The Mobile Wine Agent: Pairing Wine with the Social Semantic Web. In: 2nd Social Data on the Web Workshop (2009).
22. Okkyung Choi, Kangseok Kim, Duksang Wang, Hongjin Yeh and M. Hong. Personalized Mobile Information Retrieval System. Int. Journal of Advanced Robotic Systems (2012)
23. Bouidghaghen O., Tamine L.,Boughanem M.. Personalizing Mobile Web Search for Location Sensitive Queries. 12th IEEE Int. Conference on Mobile Data Management (2011)
24. M. Arias, J. M. Cantera, J. Vegas: Context-Based Personalization for Mobile Web Search, Very Large Data Bases Conference, 23. - 28. Auckland, New Zealand (2008)
25. V. Woensel, S. Casteleyn and O. Troyer. Applying semantic web technology in a mobile setting: the person matcher," Web Engineering, LNCS 6189, pp. 507- 510,Austria (2010)
26. Keßler C., Raubal M.,Wosniok, C. Semantic rules for context-aware geographical information retrieval. Europ. Conf. on Smart Sensing and Context (EuroSSC). LNCS 5741,p.p 77–92 (2009)
27. C. Venezia, C. A. Licciardi and A. Salmeri. Rule based dynamic adaptation of mobile services based on context. ICIN (2008)
28. K. Bayrhammer, T. Grechenig, H. Köstinger, M. Fiedler, W. Schramm.Using a Rule-Based Service Bus for Controlling Dataflows in Patient Centric Hospital Information Systems. In Proc. of the AMA IEEE Medical Technology Conference 2011, IEEE (2011)
29. Duane K. FieldsMark A. KolbShawn Bayern. Web Development with Java Server Pages. Manning Publications ISBN:193011012X (2001)
30. Broekstra J., Kampman A., and van Harmelen F. Sesame: An architecture for storing and querying RDF data and schema information. In H. Lieberman D. Fensel, J. Hendler and W. Wahlster, editors, Semantics for the WWW. MIT Press, (2001)
31. Horridge M, Bechhofer S.The OWL API: A Java API for Working with OWL 2 Ontologies. OWLED 2009,6th OWL Experienced and Directions Workshop, Chantilly,(2009)
32. Yuh-Jong Hu, Ching-Long Y., L. Wolfgang. Challenges for Rule Systems on the Web. RuleML '09 International Symposium on Rule Interchange and Applications (2009)
33. Paschke A., Kozlenkov A., H. Boley. A Homogenous Reaction Rule Language for Complex Event Processing, 2nd International Workshop on Event Drive Architecture and Event Processing Systems (EDA-PS 2007), Vienna, Austria, (2007)
34. H. Boley, S. Tabet and G.Wagner. Design Rationale of RuleML: A Markup Language for Semantic Web Rules, Proc. SWWS'01, Stanford, (July/August 2001)
35. David Hirtle. TRANSLATOR: A TRANSlator from LAnguage TO Rules. In Proceedings of the Canadian Symposium on Text Analysis (CaSTA),(2006)
36. Christian de Sainte Marie, Gary Hallmark, Adrian Paschke. RIF Production Rule Dialect (2nd Edition), W3C Recommendation 5 February 2013, *http://www.w3.org/TR/rif-prd/*
37. S. Liang, P. Fodor, H. Wan, M. Kifer. OpenRuleBench:An Analysis of the Performance of Rule Engines. WWW 2009 Madrid (2009)
38. Ernest Friedman-Hill: Jess in Action.Rule-Based Systems in Java. Manning Publications. ISBN-10: 1930110898, pp. 32-33(2003)
39. A. Giurca, M. Tylkowski, M. Muller. RuleTheWeb!: Rule-based Adaptive User Experience, RuleML2012@ECAI Challenge, 6th Int. Symposium on Rules, Montpellier (2012)