Expert Systems with Applications 42 (2015) 4443–4459

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A context-aware web-mapping system for group-targeted offers using semantic technologies

Iosif Viktoratos [a,*], Athanasios Tsadiras [a], Nick Bassiliades [b]

[a] Department of Economics, Aristotle University of Thessaloniki, GR-54124 Thessaloniki, Greece
[b] Department of Informatics, Aristotle University of Thessaloniki, GR-54124 Thessaloniki, Greece

ARTICLE INFO

Article history:
Available online 31 January 2015

Keywords:
Rules
Ontologies
Semantic web
Location Based Services
Context
Points of interest
Group-targeted offers

ABSTRACT

Existing location based systems that propose offers to their users do not provide points of interest (POIs) owners with the capability to flexibly customize their target groups of people/customers based on their context but they simply rely on the pre-determined application's methods to approach them. These systems also suffer from information overload, often providing offers to a user that are neither valid nor interesting because they do not match his/her context. Moreover, these offering strategies are not interoperable among different systems. In this paper, we present the design and implementation of an innovative web-mapping context-aware system called "SPLIS" (Semantic Personalized Location Information System) that utilizes Semantic Web technologies for delivering group-targeted offers from POI owners to users/potential customers. The presented system (a) adopts the schema.org ontology, (b) uses RuleML-compatible rules to represent group-targeted POI offers, (c) combines at run-time the above to match user context with suitable offers, and finally, (d) visualizes offers in an intuitive way. The paper also reports on a user evaluation of the system.

1. Introduction

Nowadays, mobile commerce applications play an important role in everyday life for millions of people and they have changed the way we do things in many sectors such as financial transactions, navigation, marketing, entertainment, etc. (Baldauf, Frohlich, Masuch, & Grechenig, 2011). A popular subsector of m-commerce is Location Based Services (LBS), which are services that utilize user location (Michael & Michael, 2011). LBSs should offer rich and personalized content to users and provide efficient answers tailored to their needs and preferences (Hand, Cardiff, Magee, & Doody, 2006). If we take as examples (a) a LBS user who is driving and looks for a coffee shop close to him, or (b) a user who is looking for transport, presumably the first user requires promotions or offers provided by cafes while the second requires very different kind of information such as suggestions for the nearest taxi stand, transport office, or car rental promotions. These examples make clear that successful LBSs should be capable of offering emerging information retrieval and direct information to users, relevant to their situation. Every piece of information which is used to characterize the respective situation (known as context) leads to this direction because user requirements are closely related to the user profile (e.g. preferences, social state, etc.) and the environment (place, time, weather, etc.) (Dey & Abowd, 2000; Emmanouilidis, Koutsiamanis, & Tasidou, 2012; Kwon & Kim, 2012; Kwon, Yoo, & Suh, 2006). Context-awareness has been associated with LBSs and the scientific community and the industry focus on collecting, utilizing and interpreting contextual knowledge by developing relevant hardware technologies (e.g. GPS) and software such as semantic technologies (e.g. ontologies, rules).

Concerning the second domain referred above, semantic web standards such as RDF/S and OWL, usually referred as ontologies, enhanced contextual knowledge collection and perception process because (Bizer, Tom, & Tim, 2009; Eberhart, 2003; Her et al., 2010; Ilarri, Lllarramendi, Mena, & Sheth, 2011; Kim & Jin, 2010; Kim, Suh, & Yoo, 2007; Lee et al., 2005):

(a) They offer the ability to represent the structure of physical entities and the associations between them (e.g. representing concepts such as user profiles, places, etc.).
(b) They enable knowledge sharing, semantic interoperability, through reasoning, and seamless communication between heterogeneous systems, by providing a formal and general knowledge representation and reasoning standard.
(c) They can be reused and extended easily, saving a lot of time and effort for developers.

* Corresponding author.
E-mail addresses: viktorat@auth.gr (I. Viktoratos), tsadiras@auth.gr (A. Tsadiras), nbassili@auth.gr (N. Bassiliades).

http://dx.doi.org/10.1016/j.eswa.2015.01.039
0957-4174/

Flexible context adaptation needed in LBSs can be effectively represented and enforced by combining ontologies with rules. Such examples are OWL 2 RL (Motik et al., 2012) and SRWL (Horrocks et al., 2004). Rule-based systems are more autonomous due to the following reasons (Lassila, 2005; Patkos, Bikakis, Antoniou, Plexousakis, & Papadopouli, 2007; Wu, Chang, Ho, & Chao, 2008):

(a) They are capable of conceiving context changes and respond accordingly without user intervention.
(b) They are more proactive and have the capability to offer services in advance.

The aim of our work is to combine semantics with location information services to deliver contextualized offers to users. The majority of LBSs that propose offers to their users, do not give POI owners, who are also offer providers, the opportunity to flexibly customize their target group of potential customers and its context. POI owners place their offers, and after that, the recommendation system that each developer has implemented handles them in a way determined only by the system itself. Even when the advertiser is able to customize some demographic parameters of the targeted potential customers, as in e.g. Facebook, the customization is limited to few pre-defined profile attributes, such as age, gender, etc. As a result, POI owners rely on application developers and the personalization system they have implemented and, consequently, their marketing strategy is not as effective as it could be. Besides, customers often get irritated by information overload (e.g. offers that exist but do not apply to them). Therefore, real-time contextualization of both content and offers is necessary and prevents from degrading user experience.

For this purpose, we have implemented a system called "Semantic Personalized Location Information System – SPLIS[1]", which provides POI owners with the ability to build more accurate and more profitable relationships with their potential customers, as their offers will be much more specific and related to them (A preliminary implementation of the system has been presented by Viktoratos, Tsadiras, and Bassiliades (2012)). To achieve the above, SPLIS:

(a) Collects data concerning POIs from Google Places API.
(b) Adopts a widely used ontology (schema.org) to represent persons, POIs and their relations. POI owners can enrich the schema at run time by adding their own properties.
(c) Provides POI owners a form-based web interface to deploy their offers according to their policy, regarding the appropriate target group and the context of each user (user's profile, place, time, weather, etc.).
(d) Transforms these offers into machine understandable rules. RuleML format is used in order these rules to be shared with other systems in the web. Jess translation is employed to make them machine executable.
(e) Stores metadata and rules in the form of RDF triples (using Sesame) for knowledge sharing and reusability.
(f) Displays personalized information on Google Maps[2] to regular users/potential customers in order to quickly find a place or an offer matching their profile.

SPLIS not only possesses the advantages of ontologies and rules which are discussed above but it also overcomes some of the disadvantages of rule-based systems by adopting a dynamic knowledge-based approach (allowing POI owners to add data and rules at run time). This capability will be discussed in detail in Section 2, compared with other approaches. In Section 3, the design of SPLIS system is discussed, while Section 4 describes implementation details. In Section 5 the system operation process is illustrated. In Section 6 the functionally of the system is exhibited by the use of a number of examples. The results of the evaluation of the SPLIS system by end users are presented in Section 7. Finally, Section 8 presents the conclusions of our work and discusses future directions.

2. Related work, motivation and contribution

Section 2.1 below presents (a) an overview of services that exploit semantic web technologies to provide high level personalization and inspired our work, and (b) an overview of services that provide personalized offers and promotions. Section 2.2 presents some drawbacks of the works that are discussed in Section 2.1 and explicates SPLIS overall contribution.

2.1. Personalization in LBSs and LBSs which provide nearby offers

Many researchers have been studying the provision of personalized POI's, services or tasks to users. Noguera, Barranco, Segura, and Luis (2012) used recommendation techniques such as collaborative and knowledge based filtering to propose a hybrid recommender system, which is combined with smartphones visualization capabilities (e.g. 3D) to achieve better presentation of information. Furthermore, Liu, Liu, Aberer, and Miao (2013) proposed a model for category-aware POI recommendation. They used Matrix factorization to predict a user's preference transitions over categories and after that his/her preference on locations in the corresponding categories. Liu and Aberer (2013) also in SoCo combined contextual information and social network information to improve the accuracy of recommendations. Yuan, Cong, Ma, Sun, and Thalmann (2013) defined a new problem of time-aware POI recommendation as an extension of the conventional POI recommendation problem by considering the temporal influence in user activities. In the same spirit, Cheng, Yang, Lyu, and King (2013) approach the problem of POI recommendation by proposing a novel matrix factorization method which includes personalized Markov chains and localized regions.

Liagouris et al. (2011) also implemented a mobile travel guide by referring to the concept of task computing. A general platform is supported, so that users can select tasks defined from a related ontology and access relevant content. Similarly, Ciaramella, Cimino, Lazzerini, and Marcelloni (2009) proposed a situation-aware service recommender. Rules in SWRL format are used to determine each condition and, after that, a set of related tasks is recommended. Another service designed for tourists is COMPASS (Van Setten, Pokraev, & Koolwaaij, 2004), which, through an interactive web interface, displays personalized information based on user preferences and contextual attributes (e.g. last time user visited a place). Hawalah and Fasli (2014) implemented a system that can track user interests and build contextual ontological profiles to provide personalized recommendations. In the same spirit, Guo and Lu (2014) proposed a novel contextual information recommendation model based on distributed cognition theory after modeling user interest information structure. Another interesting service is that implemented by Bozzon et al. (2011), which recommends to the users combinations of nearby POI's by ranking available results.

Additionally, Keßler, Raubal, and Wosniok (2009) combined data collected from sensors with ontologies and rules in SWRL format for utilizing complex context information and provide personalized recommendations for surf spots. Another application of this area is Sem-Fit (García-Crespo et al., 2011), which uses fuzzy rules

[1] Can be accessed at http://tinyurl.com/splis-login
[2] http://maps.google.com

to recommend hotels to a user. The customer provides an evaluation of the returning results and the rules are updated so as to provide better results. Several researchers make use of social media data such as foaf, Facebook, twitter, etc. to achieve better personalization. For instance, Serrano, Hervás, and Bravo (2011) proposed a tourist information service which combines RDF data taken from sources such as foaf profile with rules in SWRL format to recommend places of interest related to user profile. Patton and McGuinness (2009) also implemented a mobile agent for wine and food recommendation. The agent is being integrated with social media such as Facebook and twitter. Data from these sources are stored in RDF triples and retrieved for personalized recommendations. Woensel, Casteleyn, and Troyer (2010) proposed a person matching application based on data retrieved from a FOAF profile and identification techniques such as RFID, Bluetooth, etc. PhotoMap (Viana, Filho, Gensel, Villanova-Oliver, & Martin, 2008) utilizes rules in SWRL format to attach physical and social context to photo shots (e.g. where the photo was taken and who was there). Furno and Zimeo (2014) implemented a service for smart devices to propose a model for context aware service composition by extending the OWL-S ontology. Skillen et al. (2014) also used SWRL rules to enhance personalization of Help-On-Demand services in pervasive environments.

Seeking for nearby offers and promotions is a popular subcategory of LBSs and there are many successful commercial applications in this area. Some of the most popular (having millions of users all over the world) are Foursquare Specials (https://foursquare.com) and Yelp (https://biz.yelp.com). Another interesting approach in this domain is that implemented by Niforatos, Karapanos, and Sioutas (2012). They proposed a platform for proactive service discovery which runs in background and informs the user about nearby offers while he/she is on the move. Ebrahimi, Villegas, Müller, and Thomo (2012) developed a deal recommendation system called "Smarterdeals" that utilizes users' changing personal context information to deliver highly relevant offers. SmarterDeals relies on semantic technologies and recommendation algorithms based on collaborative filtering.

2.2. Motivation and scientific contribution of our work

In respect to the LBS systems that were presented in Section 2.1, the following issues (according to our work and others as e.g. Dell'Aglio et al., 2010) can be reported:

(a) Regarding the systems that provide personalization by recommending nearby POIs, although they are accurate, they are completely based on the developers' personalization system as discussed in the introduction section. POI owners have no flexibility in attracting customers (e.g. when they have special offers or an event, specific hours/days, etc.) and rely on the developer's recommendation model. These systems also do not exploit the advantages of semantic technologies such as reasoning (e.g. POI category and its subcategories) or interoperability (e.g. data sharing).
(b) Regarding the Knowledge-based LBS systems, they use a predefined set of rules created by the developers. Apart from the fact that these rules are not enough to describe every possible situation, many system features change from time to time (e.g. users, place data updates), making system policies obsolete. Also the design, development and maintenance of new rules are time consuming processes which require a lot of developer's effort. Finally, developers' rules are not always effective for every user.

In addition to the above, regarding systems that focus on offers and deals, the following limitations can be identified:

(a) These systems do not provide POI owners with the capability to set manually (a.k.a. customize) their target group and its context. Owners can only define the time period where their offers are valid and after that they rely on the system's personalization methods/algorithms.
(b) They suffer from information overload, providing offers to a user that are neither valid nor interesting for him/her, since they do not match his/her profile and/or context.
(c) Their offers are not interoperable among different systems or web services. Some web-based systems/services provide Restful APIs for third-party applications but they give a limited number of offers or even charge after a number of calls.

The aim of our work is not to provide personalization by recommending POI and offers (personalized offers is only one aspect of our work) but to design a novel and completely different methodology/approach and prove its worthiness by developing a system that overcomes the issues presented above. The proposed methodology and system (called SPLIS) uses (a) a well known ontology such as schema.org (for interoperability purposes) and (b) web-exchangeable rules (for contextualization/customization/interoperability purposes), to address these issues in the following ways:

(a) The system is fully compatible with a popular ontology such as schema.org (also adopted by Google, Bing and Yahoo), incorporating and extending dynamically its RDF Schema version. Schema.org ontology offers the capability to represent a wide range of physical and digital entities (persons, places, movies, etc.), and also the associations between them. A consistently used and widely accepted ontology such as the above is crucial for the acceptance of the system by end users and the reusability of its data and knowledge.
(b) The system enables a fully dynamic rule base (POI owners can add rules at run time). It takes advantage of people collaboration and engages non technical users (POI owners in our case) to take part in the knowledge construction process. This makes SPLIS more and more intelligent as soon as more rules are added to the system. Furthermore, as soon as POI owners take part in the knowledge construction process, their rules are more knowledgeable, intuitive and, hence, effective than those provided by system developers and they can reach their audience more efficiently.
(c) The system provides POI owners with full customization concerning their data and their target group in order to assist them to add to the system their specific group-targeted offering policy.
(d) The system performs real-time contextualization of content and offers in order to deal with information overload. Valid offers (i.e. offers which match user's context, according to the rules of POI owners) are dynamically identified and highlighted to the end-user.
(e) The system constitutes a framework for POI owners to communicate their offers to various systems/services across the web. The rules that implement the specific group-targeted offer policy of the POI can be placed even to the web site of the POI and they can be extracted through crawling by our system or others.

The main scientific contribution of our work can be summarized in the following. We employ a user-defined, knowledge-based offering policy, based on logical rules. This is the first time rules are used in such a way in a location-based information system. The advantages of using rules are that rules are declarative so POI owners can easily grasp them, define them, customize them and maintain them. Furthermore, rule languages offer advanced expressiveness in combining and matching information from various data sources such as the device of the user (location/time), the profile of the user (preferences, demographic info) and the profile of the POI (location, offering policies). Thus, POI owners have a great flexibility in expressing either fine-grained or coarse-grained offerings, according to their own choice. Finally, rules can easily provide explanation to the end-user why an offer is valid or not for him/her. Hence, the system can earn the trust of the end-user more easily. All the above are coupled with Semantic Web technologies in order to increase the interoperability of the system with other systems or web information sources. For example, data about POIs can be found either at Google Maps or in the web page of the place, provided the RDFa extension to HTML is used to semantically annotate web pages. Once the data are collected in our system they be easily exported to other uses and/or they can be further enriched by linking them to other web data sources through the Linked Open Data method. The same interoperability claim is valid also for our system's offering policies, which are expressed in the RuleML language, a de facto web rule interchange language standard. Since rule conditions and conclusions refer to the collected RDF data, both rules and data can be exported and re-used for other purposes.

3. Design and general idea

In everyday life, many businesses, usually adopt a rule-based offering policy in order to set up their explicit business or marketing strategy. For example, a coffee shop offers 10% discount to students every Friday afternoon, a museum offers a free entrance to children under 12 years old on weekends, etc. On the other hand, LBS users search for such kind of information (e.g. special offers) 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 his/her favorite food. It is apparent that customers are more willing to accept an offer if they are already interested in it (Li & Jhang-Li, 2009). For example, if someone searches for a place to eat, an offer from an Italian restaurant is suitable and, consequently, efficient. SPLIS aims to provide a direct interaction platform between service providers (POI owners in our case) and users-potential customers, combining customer needs with highly targeted personalized information. On the one hand, owners can expose their marketing strategy by adding meta-data related to their services/products (meta-data based on the schema.org ontology, or even custom meta-data, by extending the schema.org ontology) and by authoring rules that represent their group-targeted offers. On the other hand, regular users/potential customers enjoy proactive contextualized information.

Every time a user logs into the system to search for a point of interest, SPLIS gets user's context, evaluates the rules associated with nearby POIs and delivers personalized offers to the user, depending on the rules that have fired. These offers are then visualized on Google Maps. SPLIS is able to handle rules that involve:

(a) every existing property of a POI,
(b) user's occupation (e.g. a coffee shop offers discount to students),
(c) user's hobby (e.g. a bookstore has an offer for a user who likes reading),
(d) user's gender (e.g. a pub has special prices for women),
(e) user's age (e.g. free entrance for children),
(f) user's location (e.g. a coffee shop decreases coffee price for users who are less than 400 meters away),
(g) time (a restaurant is closed on Sundays or a coffee shop have special prices between 17:00–19:00),
(h) weather (e.g. a discount to ice-cream price if weather is cloudy).

4. SPLIS architectural design

The architectural design of SPLIS is presented in the UML component diagram of Fig. 1. The core component of the proposed system is Sesame, which supports multiple operations such as storing, querying and inferring large sets of RDF triples (Broekstra, Kampman, & van Harmelen, 2001). Sesame can easily be embedded into java-based applications (http://www.openRDF.org/).

An important part of the system is the rule representation process because human understandable policies have to be translated into a computer understandable language, in order to be executed by an information service. In our case, RuleML and being more specific, Reaction RuleML (a clone of RuleML) was adopted (Hu, Ching-Long, & Wolfgang, 2009; Paschke, Kozlenkov, & Boley, 2007). Reaction RuleML was chosen because it can represent production (Condition Action) rules, which is the type of rules that was needed to represent POI owners' rules (Boley, Tabet, & Wagner, 2001; Grosof, 2003). As an XML-based language it provides rule interoperability among different systems on the web and is close to natural language (Hirtle, 2006; Kontopoulos, Bassiliades, & Antoniou, 2008).

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, when e.g. checking the profile of a user or a POI, in order to decide whether an offer is relevant or not. RIF-PRD (de Sainte Marie, Hallmark, & Paschke, 2013) could also be a candidate, currently not supported by tools as much as RuleML.

In order to build a rule-based system like SPLIS, a rule representation language has to be translated into a machine executable language. Rule engines such as Jess, Drools, Prova (Liang, Fodor, Wan, & Kifer, 2009) are needed for this purpose. Jess was chosen to implement the core of SPLIS, because of the fact that it is a lightweight rule engine that connects well with web technologies, which were needed for SPLIS system implementation (Friedman-Hill, 2003). Jess is based on an improved version of the Rete algorithm and uses adaptive indexing techniques to improve performance (full indexing can be too expensive). Liang et al. (2009) performed a detailed evaluation about the performance and the scalability for several rule engines, regarding different problem sets. The results signified that Jess is one of the best choices regarding our case (e.g. type of data, type of rules, etc.). RuleML rules are transformed to Jess rules by using XSLT files (Sherman, 2007).

Common client-oriented web technologies such as HTML, JavaScript and AJAX, were used for visualization. Java Server Pages (JSP) technology was chosen as server side component. It is an example

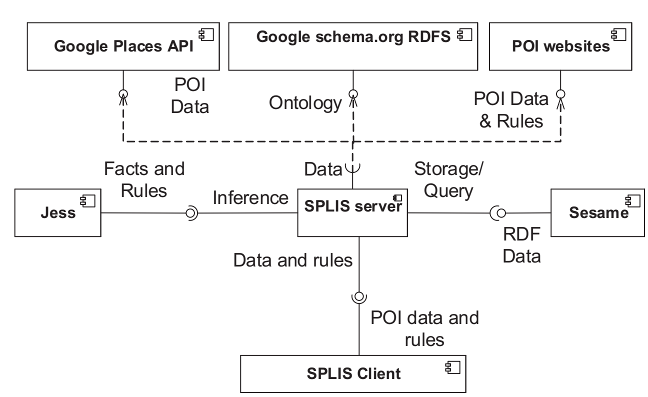


Fig. 1. SPLIS architectural design.

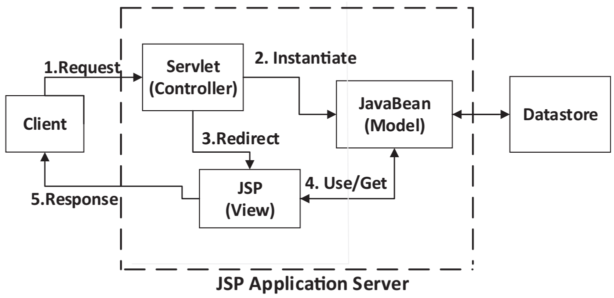


Fig. 2. SPLIS JSP server internal architecture.

of Model 2 architecture (Fields, Kolb, & Bayern, 2001). Fig. 2 illustrates the internal architecture of the SPLIS JSP application server. JSP pages contain Java combined with HTML, separating the data modeling logic from the data display logic, in order to make the User Interface more flexible (i.e. using dynamic database queries). JSP is a high-level abstraction of Java servlets; it is translated into Java servlet before being run and it processes HTTP requests and generates responses like any other servlet (Duane et al., 2001). It could be easily integrated with the vast majority of available java-based rule engines and with technologies used for RDF, RDF/S and OWL storage and querying such as Sesame, Jena, Protégé, etc (Broekstra et al., 2001; Horridge & Bechhofer, 2009).

A lightweight mobile version for Android devices has also been implemented. Being more specific, the SPLIS mobile application supports all functionalities concerning regular run time user operations (for example search for POIs, rate them, like, etc.). These operations are described in detail in the following section. SPLIS users can access it either by (a) web browser from PC, mobile, etc.[3] or (b) by installing the mobile application to their smartphone[4].

5. SPLIS operation process

In this section, SPLIS operations are fully described. It includes the (a) Data discovery and collection, (b) Information presentation and (c) User operations. The UML sequence diagram (Fig. 3) displays the processes that have to be completed in order these operations to be executed.

5.1. Data discovery and collection

First of all, the system collects all the necessary data from external sources. This operation is responsible for gathering and adding new data into the system repository, related to users and places. 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, namely the "Person" class of schema.org. User profile property values are stored in the triple store. After completing the registration process, the user is able to log into the system by entering his username and password. If authentication operation is completed successfully, a process concerning POI data collection begins. Combining multiple sources for dynamic data discovery is one of the most important functionalities, as new data appear constantly on the web. Three steps are included:

(a) **Ontology loading**. Initially the ontology is loaded into the Sesame repository so that the system to be up-to-date with the latest official updates (e.g. new properties) of the selected ontology.
(b) **Data update**. After ontology loading, the system obtains user's position (via GPS in a mobile device or via the IP address in a desktop/laptop PC) and retrieves the nearest POIs (so as to reduce computational cost) from Google Places API. A mapping between the schema.org RDF/S ontology classes/properties and API categories/attributes is being performed automatically for compatibility. Specifically, every category from API was matched manually with a related schema.org class based on similarity (e.g. API "lodging" is matched with schema.org "LodgingBusiness", etc.) and every attribute which is being parsed from the API with the related property (e.g. name with http://schema.org/name). Afterwards, the mapping scheme is embedded into the application[5]. Data from Google Places API are stored into the RDF repository. If a POI already exists in the system, SPLIS updates its related data with the latest information. For update process optimization, SPLIS compares existing data with the new one and updates them only if there is a difference.
(c) **Crawl websites**. A crawling mechanism has been implemented to retrieve new data from web sources such as POI websites. SPLIS receives website URLs from nearby places and searches for microdata compatible with schema.org ontology (the method of annotating a web page with properties and values in microdata format, compatible with schema.org, is described in detail in schema.org's official webpage). SPLIS collects those data (properties and their values) and stores it to the repository as RDF triples. The system is also capable of handling rules, by parsing a link pointing to a RuleML file that contains POIs offer. POI owners can annotate their webpage with a property called "policy" and set as its value a link pointing to a RuleML file. Although only developers who are familiar with RuleML are able to implement this task, this functionality is supported so as to illustrate SPLIS vision to discover and interpret rules dynamically, at run time. SPLIS can parse and interpret rules dynamically as soon as they are compatible with open standards. If POI owners have difficulties in authoring such rules, they can use the rule authoring tool that exists in SPLIS, which is described in detail in the next section. A detailed example concerning this functionality will be presented in the next section.

5.2. Information presentation

After the above, the information is presented to the end user according to the rules that have fired matching his/her profile. The following steps are included:

(a) **Data retrieval**. After completing the data collection process, the SPLIS server gets the current user profile data from the repository, and along with contextual property values (location, time, day and month), builds his/her context. Afterwards, for every POI, existing data such as property values and rules (if any) are being fetched by SPLIS server.
(b) **Rule evaluation**. Data mentioned above are asserted to the working memory of the Jess production rule system. Jess evaluates rules using the asserted facts and updates POIs property values according to the rules that fired. Jess checks

[3] Can be accessed at http://tinyurl.com/splis-login
[4] Can be downloaded at http://tinyurl.com/SplisMobile
[5] The detailed mapping is available at http://tinyurl.com/SPLISappA

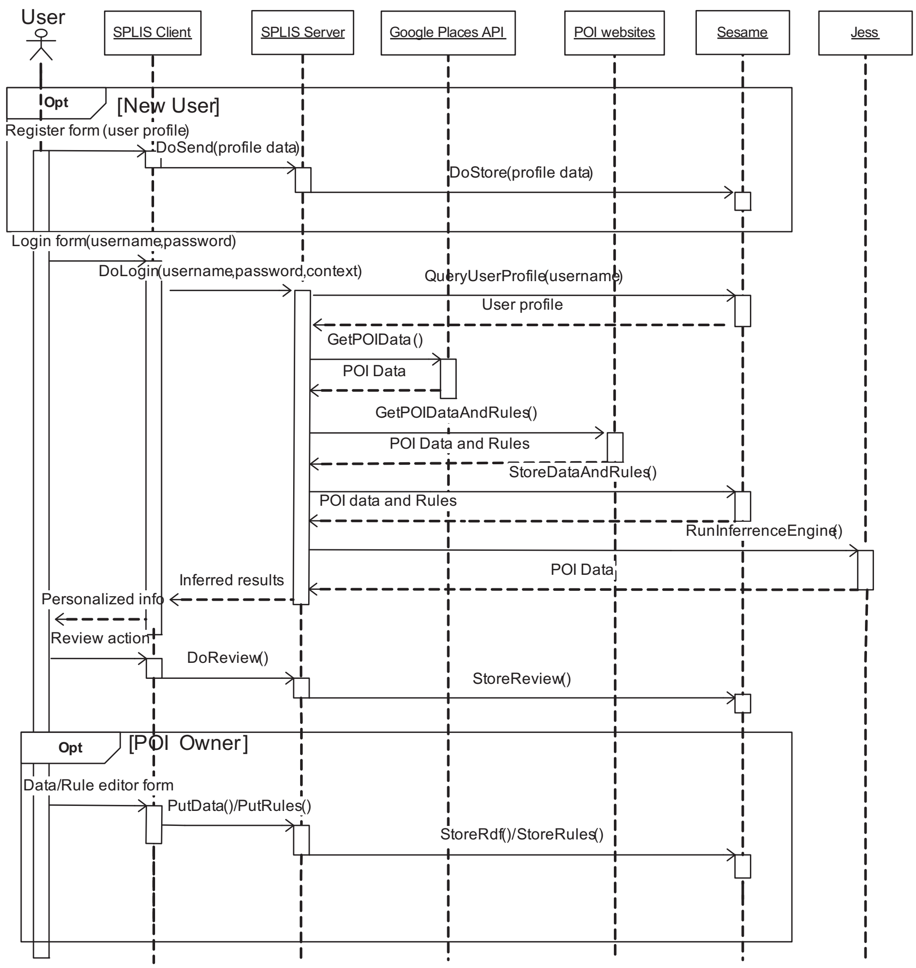


Fig. 3. SPLIS operation process.

the conditions of a rule and concludes whether or not the values of the properties involved in the RHS (THEN) part have to be modified. The new derived information is fetched by SPLIS server for presentation.

Concerning the computational complexity and the execution time, since JESS uses the RETE algorithm (Forgy, 1991) for matching rules, it is O(RFP), where R is the number of rules in the rulebase, F is the number of facts in the factbase, and P is the average number of conditions in the condition of the rules. In our case the number of rules R is kept small, since we translate and run in Jess only the offering rules about nearby POIs. Moreover, for each rule in the IF part we use only the relevant facts (6 facts about user context for the currently logged-in user) and therefore the computational complexity is kept very low.

(c) **Presentation of personalized information**. Finally, data transfer to the client is performed for presentation of personalized offers on Google map. A user-friendly interface has been implemented so that the user is able to comprehend easily the general idea and find quickly an offer associated to him/her. First of all, different markers are used for better illustration. In detail, except from the standard red marker for POI representation, (a) a yellow marker indicates that the place contains a rule base of targeted offers but no rule fired for the current user, (b) a green marker indicates that the place contains a rule base and at least one rule has fired for the current user, and (c) a crown over the marker indicates that the current user is the owner of the specific POI (so the user is able to manage its associated rule base). By clicking on 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). The user can also execute a number of operations (described below) directly, by clicking the corresponding buttons in the available menu. Finally, it's worth mentioning that in the mobile version (where the user's environment changes rapidly) SPLIS: (a) gets only 15 nearby POIs to reduce the response time (sorted by POIs that contain a group targeted offer), and (b) updates user's position every 1 min, being able to monitor user context changes.

5.3. User operations

User operations are related to two different types of users, (a) a regular user/potential customer and (b) a user who owns a POI and is able to modify its meta-data and offer policies.

5.3.1. Regular user operations

A regular user, apart from seeking for information, can directly interact with the available POI's. First of all, he/she can contribute by writing a review, rating them, or adding "likes". In other words, users do not only provide useful information to other people but they also create information that can be used during rule creation or rule evaluation (e.g. place average rating). They are also able to report abuse (e.g. fake offers) for security and reliability, by making a negative review. A user is also able to insert his/her own POI's (if any).

Apart from this, the "policy" class and its properties, such as the property "policy explanation", have been added.

5.3.2. POI owner operations

A POI owner can update the place's profile by adding new data to its properties through a user-friendly form-based interface. For correct data entry, properties (names, types) are dynamically collected from the RDF Schema depending on the place category.

Apart from asserting data, such as property values, a POI owner is able to extend the schema by adding his/her custom properties. The user can choose if these properties are related only to a specific place (in this case only the values of the newly added properties are stored as triples in the repository) or if they should be attached to a POI category and its subcategories (in this case, the property is also added to the schema). The POI owner can also choose the type of the property among text, number and date.

One of the main functionalities of SPLIS is that of adding rules. A user-friendly editor has been implemented so that users can easily add rules through completing specific forms. In more detail, they are able to customize (a) the type of user (person or a subclass), (b) any property related to the specific POI, and/or (c) any property regarding user context. Rules are transformed to RuleML, for rule exchange purposes, and after that RuleML is transformed to the Jess rule language via XSLT[6], so as to become machine executable, as discussed above. For example, Fig. 4 displays how the rule "If a person likes sports, there is a special offer on Wednesday's" is created using the web rule editor of the system.

After submitting this rule, it is transformed first into RuleML and then to JESS format, as illustrated in Table 1. Concerning the JESS representation, the JESS salience operator is used for resolving rule conflict issues, the "modify" action updates the values of the corresponding properties, the "?fact" variable is a JESS variable that is programmatically assigned from Java to the ID of the particular place and "EXPLANATION" is a Java variable that stores the explanation string in order to be presented afterwards to the end user, via Java. Concerning the class Person and its subclasses, in Jess we represent this as a value of the slot "type" in order not to create new templates for every subclass (regarding users, there is only one template that is the template person). The system gets the value regarding the type and stores apart from this and its subclasses (e.g. type = "University Member" OR "Faculty" OR "Administrative Staff" OR "Student").

Place owners are able to repeat this process and add several rules. They can also determine the rule priority in case of rule conflicts. Rule priorities are handled through the rule salience value in Jess. The same form-based interface is also provided to owners for updating the existing rules. Apart from this, unique files containing rules in RuleML format are kept to the server, in order to be available for sharing in the web. For example, POI owners after adding a rule, (apart from deleting or editing this rule) they can get either the RuleML code or a link pointing to that file. Then, they can upload it to their website or any other web link/file sharing system. Concerning security issues, the editing of the POI's rule base is authorized only to the place owner.

6. Use cases

In this section use case scenarios are presented so as SPLIS functionalities to become clearer. A POI owner's offering policy will be checked against two different user contexts.

[6] It can be accessed at http://tinyurl.com/SPLISappB

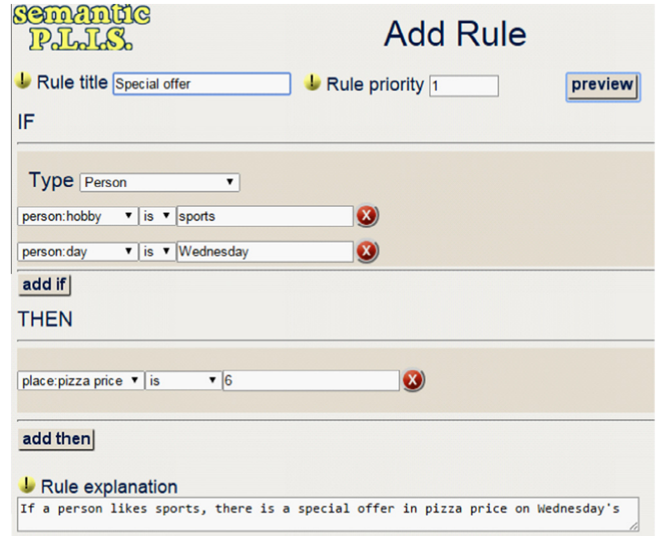


Fig. 4. An example of rule creation.

Table 1
An example of a rule in Jess and RuleML format.

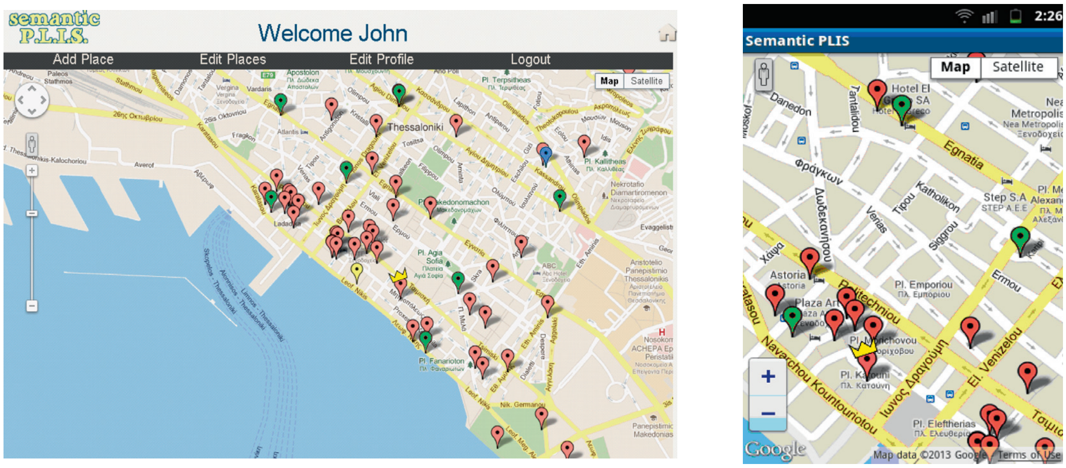
RuleML representation
⟨RuleML⟩ ⟨Assert⟩ ⟨Rule style = "active"⟩
⟨label⟩special_offer⟨/label⟩
⟨explanation⟩ If a person likes sports, there is a special offer in pizza price on Wednesday's ⟨/explanation⟩
⟨if⟩⟨And⟩
⟨Atom⟩⟨Rel⟩Person⟨/Rel⟩
⟨slot⟩ ⟨Ind⟩type⟨/Ind⟩
⟨Ind⟩person⟨/Ind⟩ ⟨/slot⟩
⟨slot⟩ ⟨Ind⟩hobby⟨/Ind⟩
⟨Ind⟩sports⟨/Ind⟩ ⟨/slot⟩
⟨slot⟩ ⟨Ind⟩day⟨/Ind⟩
⟨Ind⟩Wednesday⟨/Ind⟩ ⟨/slot⟩ ⟨/Atom⟩ ⟨/And⟩ ⟨/if⟩
⟨do⟩⟨Update⟩
⟨oid⟩ ⟨Var⟩fact⟨/Var⟩ ⟨/oid⟩
⟨Atom⟩⟨Rel⟩Place⟨/Rel⟩
⟨slot⟩ ⟨Ind⟩pizzaPrice⟨/Ind⟩
⟨Ind⟩6⟨/Ind⟩ ⟨/slot⟩ ⟨/Atom⟩
⟨/Update⟩ ⟨/do⟩
⟨/Rule⟩⟨/Assert⟩⟨/RuleML⟩
Jess representation
(defrule special_offer
(declare (salience 1))
(Person (type person) (day Wednesday) (hobby sports))
=> (modify ?fact (pizzaPrice 6))
(store EXPLANATION "If a person likes sports, there is a special offer in pizza price on Wednesday's"))

6.1. Scenarios concerning a POI owner

Two different scenarios concerning a POI owner will be presented. The first use case scenario (scenario 1a) concerns a POI owner who uses SPLIS to add data and rules, and the second (scenario 1b) demonstrates POI owner capability to add data and rules outside the system, without logging in.

6.1.1. Scenario 1a: Group-targeted rules are created and stored in SPLIS

In this scenario, user "John" is the owner of the POI "Friends Cafe". As soon as "John" logs in SPLIS, he can directly find his POI since it is displayed with a crown on top of it (Fig. 5a displays the PC browser version and 5b displays the mobile version). After



(a) Web browser-based version for User "John"

(b) mobile version for "John"

Fig. 5. SPLIS client interfaces.

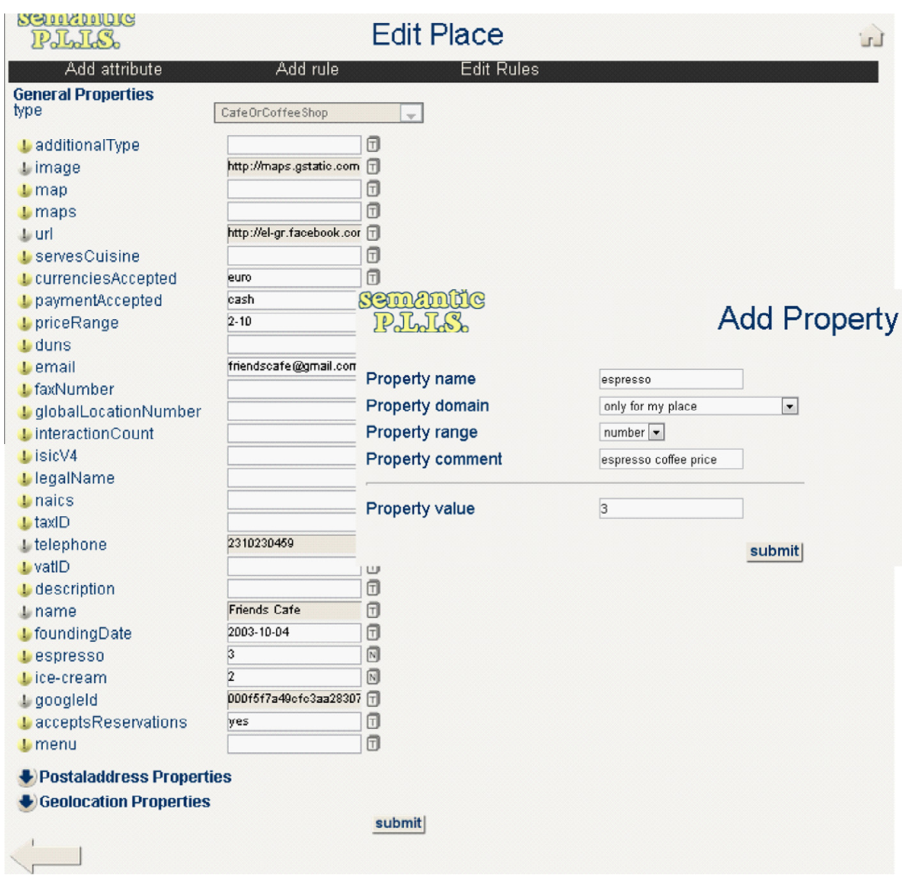


Fig. 6. Editing form of POI properties in SPLIS.

that, by choosing the "edit places" option from the menu, he can edit his place by adding/modifying data and rules. He is able to fill all properties of "Friends Cafe" by editing the corresponding form which is illustrated in Fig. 6. To assist correct data entry, property types are pulled directly from the RDF Schema and they are shown to the user next to each field of the form (e.g. [T] for text). The POI owner can click on the name of a property and in a popup box he will receive a textual comment regarding the specific property as well, fetched by schema.org. Data collected from the Google Places API are read-only and are displayed by a different color.

Let's now assume that "John" wants to add two new properties for his POI that they denote the prices of two additional product types that his shop offers, e.g. (a) the coffee-type "espresso" and (b) ice cream. This can be done by filling in all relevant information to the corresponding fields shown in Fig. 6. For example, concerning property "espresso": (a) the domain of the new property is only

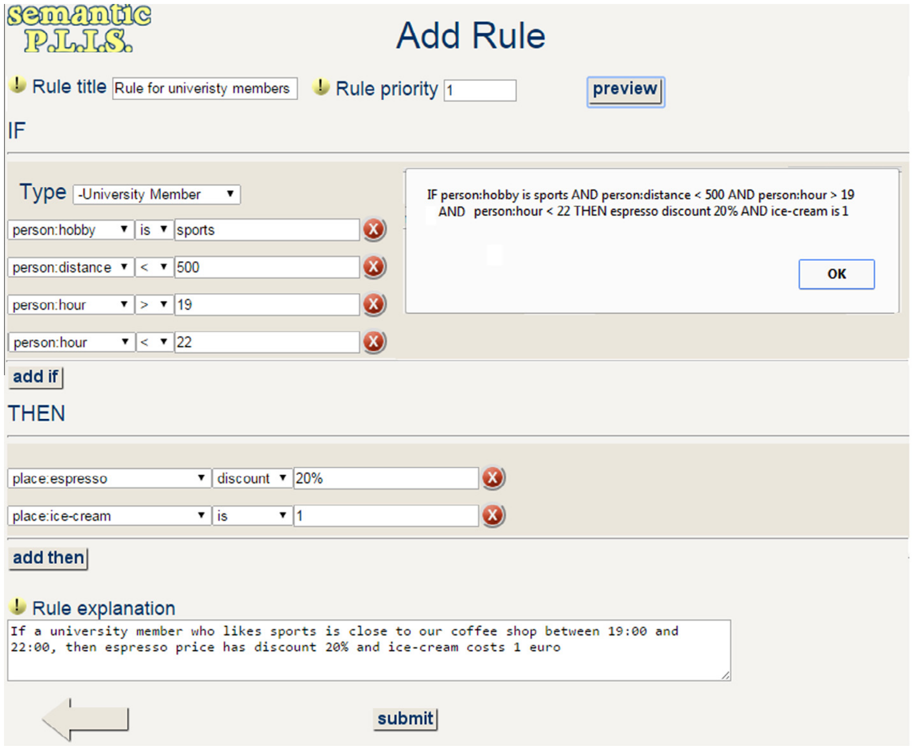


Fig. 7. The rule editor of SPLIS.

Table 2
A sample of RDF triples raw data for "Friends Cafe".

@prefix sch: <http://schema.org/>
@prefix sch-p: <http://schema.org/Place#>
sch-p:000f5f7a49cfc3aa283077c4eda2962df173fcc8 rdf:type sch:CafeOrCoffeeShop
sch-p:000f5f7a49cfc3aa283077c4eda2962df173fcc8 sch:currenciesAccepted "euro"
sch-p:000f5f7a49cfc3aa283077c4eda2962df173fcc8 sch:paymentAccepted "cash"
sch-p:000f5f7a49cfc3aa283077c4eda2962df173fcc8 sch:priceRange "2-10"
sch-p:000f5f7a49cfc3aa283077c4eda2962df173fcc8 sch:email "friendscafe@gmail.com"
sch-p:000f5f7a49cfc3aa283077c4eda2962df173fcc8 sch:url <"http://el-gr.facebook.com/pages/FRIENDS-all-day-coffee-bar/189567721082404">
sch-p:000f5f7a49cfc3aa283077c4eda2962df173fcc8 sch:acceptsReservations "yes"
sch-p:000f5f7a49cfc3aa283077c4eda2962df173fcc8 sch:streetAddress "Paleon Patron Germanou 22, Thessaloniki"
sch-p:088c0c551e8a037333dcf963e78bfdd2c21b8d2e sch:telephone "231 050 4504"
sch-p:000f5f7a49cfc3aa283077c4eda2962df173fcc8 sch:policy sch:policy92931b85-ff65-4e90-b5d2-3dbbe84952eb

the specific POI, (b) the range of the property is number, (c) a comment has been added to assist other users, and (d) its default value equals 3 (euros). Similarly, "John" adds the property "ice-cream", defining its range as a number and its default value to 2. Fig. 6 below displays all the properties along with their default values. After completing data entry, "John" is able to use the rule editor and start adding rules. Let now assume that "Friends Cafe" is a sports cafe close to the university and he would like to attract customers by inserting the rule "If a university member who likes sports is close to our coffee shop between 19:00 and 22:00, then espresso price has discount 20% and ice-cream costs 1 euro". A demonstration of the creation of this rule in SPLIS is given in Fig. 7. As it can be seen in that figure, the rule editor is horizontally split into two basic parts, the "IF" part and the "THEN" part. To begin with the "IF" part, apart from customizing the user type from a drop-down menu (Person or subclasses), John is able to customize user context by adding elements/conditions. The elements consist of a property field (place- or user-related), an operator part ("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 predefined functions for numbers; currently only the "discount" function is implemented, which calculates a discounted value for a product price) and the value field. A drop down menu was chosen so as to help user avoid mistakes. Fields for entering a title, a priority (in case of conflicts) and a textual explanation for the rule, are also provided. Also a preview of the created rule assists the user to check the rule before submitting it to the rule base. Moreover, the POI owner can click on the yellow info (exclamation mark!) icon to get a helpful comment for each field. Table 2 shows a sample of raw data concerning "Friends Café. Table 3 illustrates the data concerning the offering policy described above in RDF Turtle format. Rules are kept in RDF format only for implementation convenience (having a single database) and no processing related to RDF/RDFS is performed whatsoever.

Before going any further, it's worth mentioning that an extension to the RDF/S ontology has been made so as to support all SPLIS functionalities. The Person class has been extended by adding the subclass "University member" and its subclasses:

Table 3
RDF triples raw data for "Friends Cafe" offering policy.

@prefix sch:<http://schema.org/>		
sch:policy92931b85-ff65-4e90-b5d2-3dbbe84952eb		
sch:policy_explanation "If a university member who likes sports is close to our coffee shop between 19:00 and 22:00, then espresso price has discount 20% and ice-cream costs 1 euro"		
sch:policy_description "IF person:hobby is sports AND person:hour < 22 AND person:hour > 19 AND person:distance < 500 THEN espresso discount 20% AND ice-cream is 1"		
sch:policy_title "Rule for university members"		
sch:policy_link "http://platon.econ.auth.gr/examples/plis_new/files/Place@000f5f7a49cfc3aa283077c4eda2962df173fcc892931b85-ff65-4e90-b5d2-3dbbe84952eb.ruleml"		
sch:policy_priority "1"		
sch:Place#000f5f7a49cfc3aa283077c4eda2962df173fcc8	sch:policy	sch:policy92931b85-ff65-4e90-b5d2-3dbbe84952eb

(a) Faculty
(b) Administrative staff
(c) Student

6.1.2. Scenario 1b – Group-targeted rules are stored at POI's web site

Let us now consider an alternative scenario about user "John". In this scenario, "John" has a file that contains his shop's offering policy saved at "http://www.friendscafe.com/mypolicies.ruleml". As it was discussed in the previous section, "John" can annotate his POI's web site with metadata, including URLs pointing to policy RuleML files, as it is shown in Table 4. In this way SPLIS, or other services compatible with schema.org, could exploit these data

Table 4
Webpage annotation example.

```
⟨html⟩
...
⟨div itemscope itemtype = "http://schema.org/Place"⟩
...
⟨h4 itemprop = "ice-cream"⟩2⟨/h4⟩
⟨h4 itemprop = "espresso"⟩3⟨/h4⟩
⟨h4 itemprop = "policy"⟩http://mypolicies.ruleml⟨/h4⟩
. . ...
⟨/div⟩
...
⟨/html⟩
```

Table 5
Two user profiles used in the demonstration.

Profile						Environment		
	Name	Type	Hobby	Gender	Age	Time	Day	Location
User A	Paul	Student	Sports	Male	22	21:05	Friday	Location A
User B	Alice	Person	Cinema	Female	26	11:15	Saturday	Location B

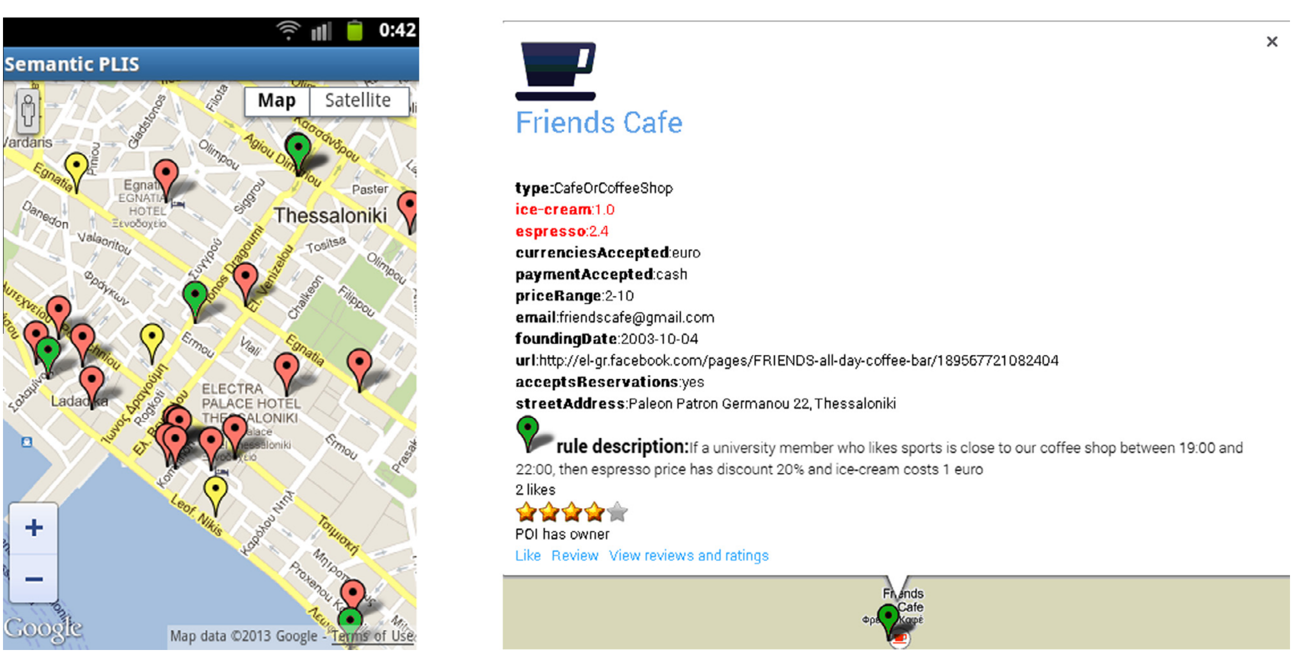


Fig. 8. (a) Mobile version of SPLIS for user A – "Paul", (b) personalized information for User A – "Paul" regarding Friends Café.

and rules. After having annotated his shop's website, there is no need for "John" to sign in SPLIS. If the URL of the "Friends Cafe" website is available at Google Places API, SPLIS parses the related information automatically. After rule authoring, POI owners are able to (a) view their offering policies in RuleML and (b) get the link (as the link presented in the 4th row of Table 3) to the published .ruleml file containing POI owner's offering policy in RuleML syntax. In this way, POI owners can provide their RuleML offering policy either in the official POI website or in SPLIS.

6.2. Scenario concerning regular users

Let us now consider the offering policy of the POI above and two different regular users in order to demonstrate SPLIS capabilities in providing personalized offers to them by matching POI policies with users' context. The scenario considers two random user profiles (see Tables 5 and 6).

(a) User A ("Paul") is a 22-year old student who likes sports, his current profile snapshot is taken on Friday, at 21:05 and his current location A is closer than 500 meters to "Friends Cafe".
(b) User B ("Alice") is a 26-year old person who likes cinema, and she has logged in the system on Saturday at 11:15 from location B, that is closer than 500 m to "Friends Cafe".

6.2.1. Scenario concerning user A

As soon as a user is logged in the system, SPLIS retrieves the user profile, evaluates rules associated with POIs "nearby" user's current location and displays personalized offers according to which rules fired and why. Taking for example User A – "Paul", the personalized offer in the mobile version of SPLIS is presented in Fig. 8a. By clicking (or touching) on yellow and green markers, "Paul" can easily see which POIs contain a rule base, which rules fired and why. For example, if "Paul" clicks on the green marker of "Friends Cafe", he is presented with all the information in Fig. 8b. Because of the fact that he is a student (a subclass of university member), he has hobby sports, time is between 19:00 and 22:00, and his current location is closer than 500 m from the cafe, the corresponding rule-offer has fired. As a result, espresso price for "Paul" has been changed to 2.4 euro and ice-cream costs 1 euro (highlighted with red color, in Fig. 8b). He also receives information about the POI data, such as the date that it was founded, the currencies it accepts, etc. "Paul" can also add a "like", a review or a rating to the POI. He can also view the reviews and ratings that have been submitted by other SPLIS users.

6.2.2. Scenario concerning user B

Similarly, when User B – "Alice" logs into the system, no rule fires for her (concerning "Friends Cafe"), because she is not a university member. As a result, this place is displayed with a yellow marker, which once again it can be clicked (or touched). In this case the information in Fig. 9 is presented. According to this, Alice is entitled for the standard espresso and ice-cream price. Notice that

Table 6
RDF triples raw data for the two users.

@prefix sch: <http://schema.org/>
@prefix sch-per: <http://schema.org/Person#>

sch-per:40	rdf:type sch:Student
sch-per:40	sch:name "Paul"
sch-per:40	sch:birthDate 1991-01-11
sch-per:40	sch:gender "male"
sch-per:40	hobby "sports"
sch-per:51	rdf:type sch:Person
sch-per:51	sch:name "Alice"
sch-per:51	sch:birthDate 1987-04-22
sch-per:51	sch:gender "female"
sch-per:51	hobby "sport"

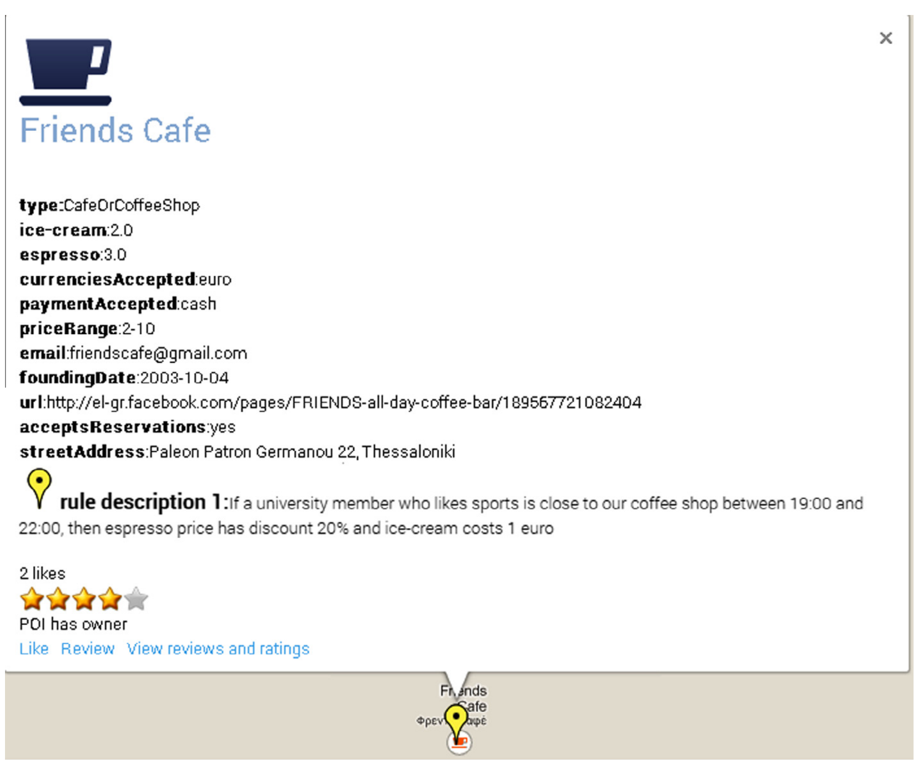


Fig. 9. Personalized information for User B – "Alice" regarding Friends Café.

Table 7
Jess execution time in milliseconds regarding the number and the type of rules.

Number of conditions in rules	POIs/rules (20 rules per POI)				
	15/300	20/400	25/500	30/600	35/700
1	47 ms	78 ms	94 ms	125 ms	156 ms
2	52 ms	91 ms	110 ms	155 ms	192 ms
3	75 ms	109 ms	140 ms	172 ms	203 ms

the rule explanation field displays all the rules concerning "Friends Cafe" so that Alice can understand (a) what kind of rules constitute the offering policy of this place and (b) the reason(s) why they were not fired for her.

7. Evaluation

To evaluate the general idea, the design, the implementation and the usability of SPLIS, an extensive evaluation regarding different aspects of the system was conducted.

7.1. Execution time performance evaluation

To begin with, a performance evaluation regarding the execution time was implemented. To achieve this, an experiment (concerning different number of POIs/rules and different types of rules) was conducted. Being more specific, we estimated the system's response time regarding:

(a) The number of POIs (15–35).
(b) The number of rules (each POI has 20 rules in our scenario).
(c) The type of rules (having 1, 2 and 3 conditions).

Regarding our server's information it possesses:

(a) Windows Server 2003 standard edition as an operating system
(b) Intel Xeon E5405 2.0 GHz, 4 cores as a processor
(c) 8 GB RAM

Table 7 and Fig. 10 below, illustrate the execution time (in milliseconds) regarding Jess. Apart from these, Fig. 11 displays the overall execution time regarding SPLIS's information presentation process. The results clearly demonstrate that even in such an extreme scenario (300–700 rules) the system performs well and the rule evaluation process is a small percentage of the overall execution time.

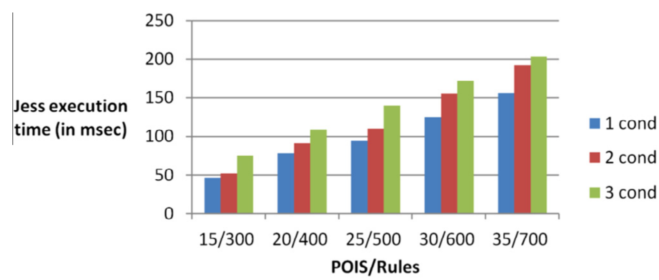


Fig. 10. Jess execution time in milliseconds regarding the number and the type of rules.

7.2. Qualitative evaluation

An electronic questionnaire (Kaplan & Maxwell, 2005; Myers, Michael, & Avison, 1997) was created and undergraduate students of Economics (age 18–22, both genders) were asked to use SPLIS and afterwards answer the questionnaire. A total of 116 replies were received. Over 67% of the participants use LBSs moderately, sufficiently or very much and over 90% of the participants consider themselves moderate, sufficient or very good users of PCs (Fig. 12). The survey consisted of three parts: The first part was related to regular run time user operations, the second with POI owner operations and the third with the system in general.

7.2.1. Regular run time user operations evaluation

After a short presentation to the system's general idea, participants were registered into SPLIS and then they logged in. Initially, they searched for places and offers and after that they answered the following questions:

Q1. Are you satisfied with the number of properties related to your profile?
Q2. How easy was to find a place which contained an offer related to your profile?
Q3. How easy was to understand why an offer matches (or not) with your profile?

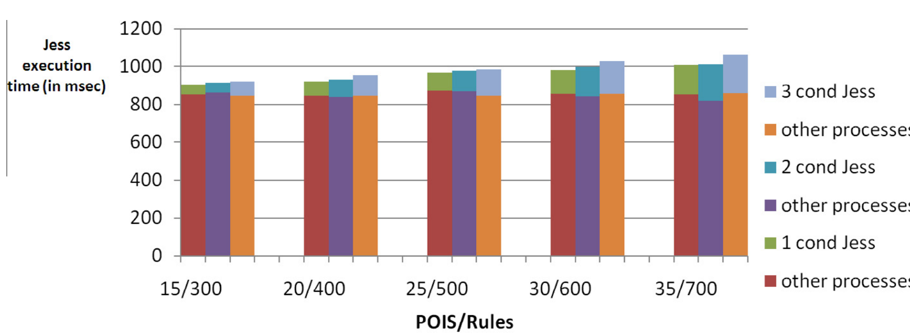


Fig. 11. Overall execution time in milliseconds, regarding the number and the type of rules.

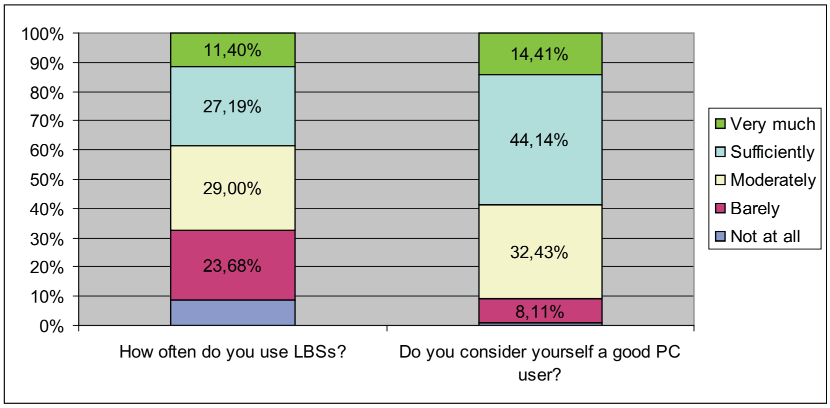


Fig. 12. Information about the participants.

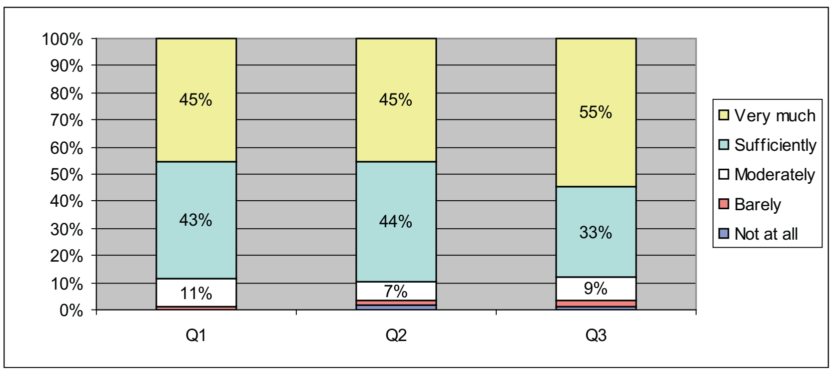


Fig. 13. Evaluation results regarding regular run time user operations satisfaction.

The results of the survey regarding the questions above are presented in Fig. 13. According to them, almost 90% of the participants were "sufficiently satisfied" or "very much satisfied" with the provided properties and they found it easy to discover a place which contained an offer matching their profile. By assigning values from 1 (not at all) to 5 (very much) to the corresponding answers, the overall average satisfaction value was calculated to 4.34.

7.2.2. POI owner operations evaluation

Afterwards, the participants took the role of a POI owner by becoming owners of a POI. They were asked (a) to modify some property values of the POI and (b) to add a new property. Then they were asked (a) to add a new rule to the POI concerning an offer and (b) afterwards to modify this rule. At the end, they were asked to answer the following questions:

Q4. Was it easy to modify properties of the POI that you own?
Q5. Are you satisfied with the number of properties related to a POI?
Q6. Was it easy to add a new property?
Q7. Was it easy to add a new rule?
Q8. Was it easy to modify a rule?
Q9. Are you satisfied with the available interface for adding rules?

The results of the survey regarding the questions above are shown in Fig. 14. According to them:

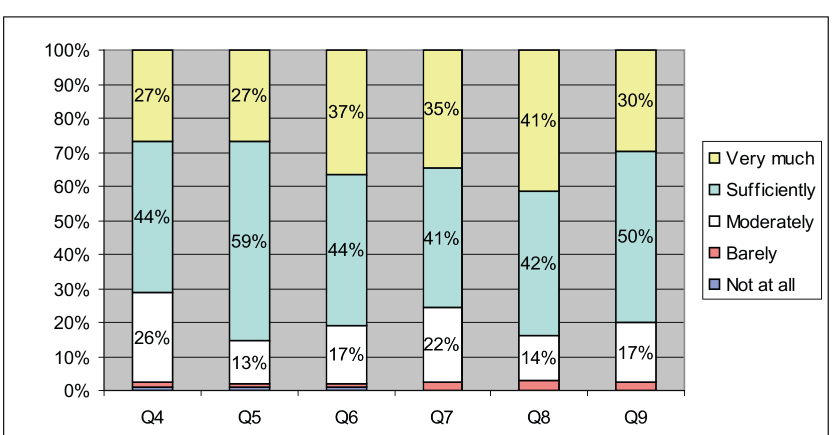


Fig. 14. POI owner operations satisfaction (questions Q4–Q9).

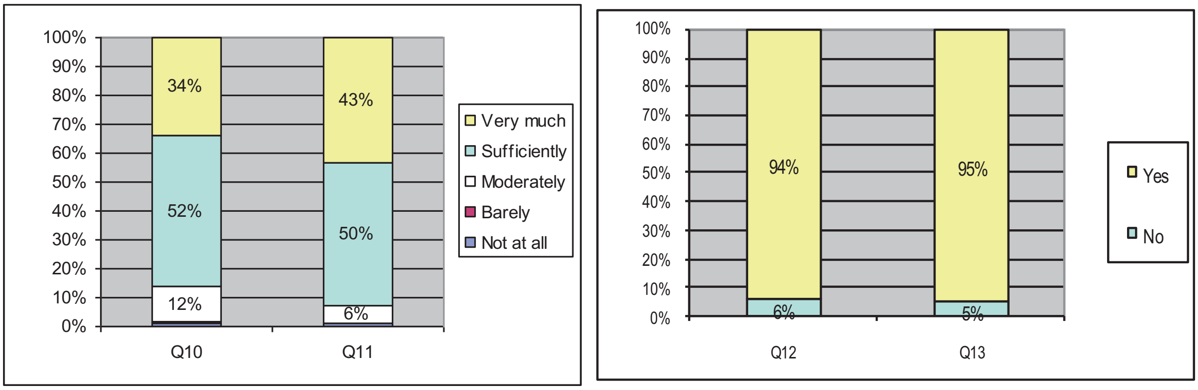


Fig. 15. System evaluation results in general (Q10–Q13).

(a) over 70% of the participants answered that is was “sufficiently easy” or “very easy” to modify place properties and add a new one,
(b) 86% of the participants were “sufficiently satisfied” or “very satisfied” with the provided properties,
(c) 76% of the participants found the task of adding a new rule, “sufficiently easy” or “very easy”,
(d) 83% of the participants found the task of modifying a rule, “sufficiently easy” or “very easy”,
(e) 80% of the participants were “sufficiently satisfied” or “very satisfied” with the available interface.

Once again, if values from 1 (not at all) to 5 (very much) are assigned to the corresponding answers, the overall average satisfaction value for POI owners operations was calculated to 4.1. Cronbach’s alpha indicator value was calculated to provide a measure of reliability. This indicator gets values between 0 and 1 and the closer it is to 1, the higher the reliability (Tavakol & Dennick, 2011). Values between 0.9 and 1 indicate an excellent consistency, values between 0.8 and 0.9 a very good consistency, etc. This indicator was calculated to 0.89 for our survey, showing a high internal consistency.

7.2.3. System usability evaluation

After completing the above tasks, participants were asked to answer the following questions related to the system in general:

Q10. How easy was site navigation?
Q11. How would you evaluate system usability?
Q12. Will you continue to use the system?
Q13. Would you recommend the system to your friends?

As illustrated in Fig. 15:

(a) 86% of the participants found site navigation “sufficiently easy” or “very easy”,
(b) 83% of the participants found that the system is “sufficiently useful” or “very useful”,
(c) 94% of the participants will continue using the system while 95% of the participants would recommend it to their friends.

7.3. Quantitative evaluation

In addition to the above, a quantitative evaluation was performed. SPLIS was compared with a stripped-down version of SPLIS which did not support offer contextualization. This approach adopted to simulate the commercial services that do not support offer contextualization (e.g. Foursquare) and inform the user by displaying a simple text message. The second version had only red and yellow markers as illustrated in Fig. 16. A red marker indicates that a place does not have any offers and a yellow marker indicates that a place has an offer, regardless if it matches user profile/context or not (the user had to click on the marker, read POI owner’s message and understand if the offer matches his/her profile/context or not). 78 pre-graduate students of departments of Economics (18–22 years old, both genders) took part. After a short introduction about the two versions of the system, they performed the following four tasks, first in SPLIS and afterwards in stripped-down SPLIS:

Task 1. Find a hotel which has an offer that matches your profile/context.
Task 2. Find a café which has an offer concerning “fredo” and this offer matches your profile/context.

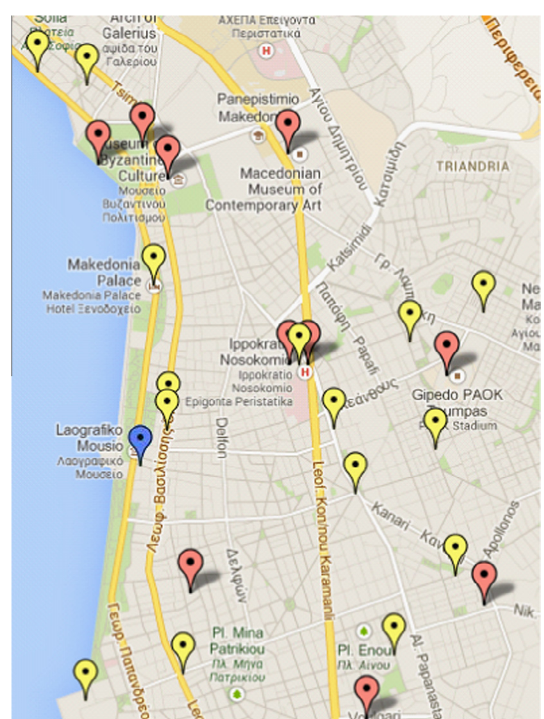


Fig. 16. A stripped-down version of SPLIS.

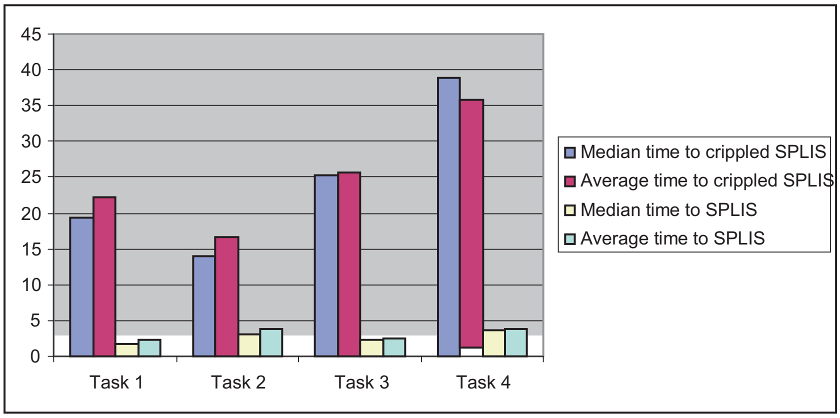


Fig. 17. Median and average time (in seconds) required for completing each task in the two versions of SPLIS.

Table 8
One-way ANOVA results.

	Sum of squares	Mean square	F	p-significance
Between approaches	939.3945	939.3945	16.25399	0.006868
Within approaches	346.7682	57.7947		

Table 9
Number of fails (time > 40 s to complete the task).

Number of fails (response time > 40 s) out of 78 attempts	Task 1	Task 2	Task 3	Task 4
Stripped-down SPLIS	11	2	4	36
SPLIS	0	0	0	0

Task 3. Find how many places contain an offer that matches your profile/context.
Task 4. Find how many places contain an offer that does not match your profile/context.

The tasks were performed in different areas on the map and had equal number of solutions so as the experiment to be fair. The available time to complete each task was 40 s and the map included 25 POIs. Regarding these POIs (a) 2 of them were solutions to task 1, (b) 1 of them was solution to task, (c) 5 of them were solutions to task 3, (d) 10 of them were solutions to task 4 and (e) 10 of them did not have any offer (they were shown with red markers).

The average and median time (in seconds) required for completing each task are presented in Fig. 17. As it can be seen, "full-functionality" SPLIS performed better in all tasks, assisting the users in completing the tasks notably sooner. In order to assess whether the differences in times are statistically significant, a one-way ANOVA test (Faraway, 2002) was conducted on the median times for completing each task (median time was used to eliminate the effect of outliers). To prove that there is statistically significant difference between the two systems, the factor p-significant of the ANOVA test should have been evaluated to a value lower than 0.05. This factor is evaluated to 0.006868 that is lower than 0.05, proving that the two systems are significantly different (Table 8). It should be noticed that time higher than 40 s was included in the results, but they were considered as fails. The numbers of fails for each task for the two systems are presented in Table 9. It can be noticed that no fail exists for the tasks performed on the "full-functional" version SPLIS.

8. Conclusions and future work

In this paper, an innovative knowledge-based LBS called SPLIS was presented, that overcomes limitations of existing LBS that propose offers and promotions. So, in contrast with existing LBS, the presented system (a) provides offers that are always valid for the user, (b) provides POI owners with the capability to customize their target groups based on their context and (c) the offering strategies are interoperable among different systems of similar functionality. SPLIS was designed and implemented to provide a connection platform between POI owners group-targeted offering policies and LBS users-potential customers. On the one hand, POI owners, by being able to specify their offering policy rules, can exhibit a highly targeted marketing strategy by reaching their right potential customers right on time. On the other hand, regular users/potential customers enjoy proactive contextualized offers through a user-friendly visual interface. The above has been achieved by SPLIS because it is an "open" system that (a) uses the schema.org ontologies for interoperability, (b) uses web based rule editor to create rules that in RuleML format, with the intention of being shared by other systems in the web, and then they are transformed to Jess rules so as to be able to be executed by the Jess inference engine, (c) stores metadata and rules in the form of RDF triples using the Sesame repository (for knowledge sharing and reusability), (d) retrieves data using Google Places API and (e) display information using Google maps.

During SPLIS evaluation process, it became apparent that by offering users the opportunity to add rules dynamically to a location-based information system, more customized and personalized user experience is provided instead of other services that do not support this functionality. Moreover, the capability of having a dynamic knowledge base (as new data and rules being added) can not only lead to powerful, autonomous and intelligent services, but also to the evolution of these services. Experimental testing, confirmed SPLIS evolution without developers intervention, as more and more users added rules to the system. The more rules were added to the system, the more interesting and intelligent it became as soon as there were rules (group targeted offers in our case) for various contexts. Besides, dynamic rule based systems such as SPLIS contain more flexible and effective rules, because of the fact that users are able to add or modify them according to their needs without developer intervention. Furthermore, the web editor facilitates rule authoring process and promotes offering policy sharing even among non familiar users. Engaging non technical users to take part in the rule authoring process will combine user-customization with the advantages of rule based systems described above. In addition to this, such policies could be easily shared and reused along the web among different users and systems. Evaluation results showed a high degree of satisfaction in (a) regular user/potential customers operations, (b) POI owner operations and (c) system usability.

Regarding the scientific contribution of our work, it should be mentioned that it is the first time logical rules are used in a location-based information system in order a user-defined, knowledge-based offering policy to be employed. This has many advantages regarding expressiveness, flexibility, reusability, explanation capabilities and maintenance. Moreover, Semantic Web technologies were applied in order the interoperability of the system with other systems or web information sources to be increased. This interoperability, even with systems of other purposes, does not apply only to data but also to rules since rules are expressed in the RuleML language, a de facto web rule interchange language standard.

SPLIS implementation can evolve in the future in various ways. For example, a way to prevent rule abuse would be the system to check at compile-time the uniqueness and the firing scope of the rule (e.g. number of times fired). So, if a rule already exists in POI knowledge base or if a rule is about to fire for every (or almost every) user, it will be rejected. Another way to extend our system is to allow users to express their contextualized personal preferences by having relevant rules. In this way, a user could have in his profile a rule such as ‘‘If today is Friday and time is after 20:00, find me some cinemas’’. Rules like the above could either be manually generated by the users themselves or they could be induced by the system by mining users' logs, likes or reviews (Liu, He, Wang, Song, & Du, 2012). Also, the above features could be integrated into a social networking framework, where user profiles, preferences and rules could be interchanged with social site profiles (e.g. from Facebook, Google+, etc.). In this case, rule recommendations could be promoted along the social graph. Furthermore, rules created by users can be shared and re-used among people that are ‘‘close’’ in the social graph, either explicitly (e.g. ‘‘friends’’) or implicitly (e.g. ‘‘similar’’ profiles).

References

Baldauf, M., Frohlich, P., Masuch, K., & Grechenig, T. (2011). Comparing viewing and filtering techniques for mobile urban exploration. *Journal of Location-Based Services, 5*, 38–57.

Bizer, C., Tom, H., & Tim, B.-L. (2009). Linked data – The story so far. *International Journal on Semantic Web and Information Systems, 5*(3), 1–22.

Boley, H., Tabet, S., & Wagner, G. (2001, July/August). Design rationale of RuleML: A markup language for semantic web rules, Proc. SWWS'01, Stanford.

Bozzon, A., Brambilla, M., Ceri, S., & Quarteroni, S. (2011). A framework for integrating, exploring, and searching location-based web data. *IEEE Internet Computing, 15*(6), 24–31.

Broekstra, J., Kampman, A., & van Harmelen, F. (2001). Sesame: An architecture for storing and querying RDF data and schema information. In H. Lieberman, D. Fensel, J. Hendler, & W. Wahlster (Eds.), *Semantics for the WWW*. MIT Press.

de Sainte Marie, C., Hallmark, G., & Paschke, A. (2013). RIF production rule dialect (2nd ed.). W3C Recommendation, 5 February 2013, http://www.w3.org/TR/rif-prd/.

Cheng, C., Yang, H., Lyu, M. R., & King, I. (2013). Where you like to go next: Successive point-of-interest recommendation. *Proceedings of the twenty-third international joint conference on artificial intelligence* (pp. 2605–2611). AAAI Press.

Ciaramella, A., Cimino, M. G., Lazzerini, B., & Marcelloni, F. (2009). Situation-aware mobile service recommendation with fuzzy logic and semantic web. *Ninth international conference on intelligent systems design and applications*. IEEE.

Dell'Aglio, D., Celino, I., & Cerizza, D. (2010). Anatomy of a semantic web-enabled recommender system. In *Proceedings of the 4th international workshop semantic matchmaking and resource retrieval in the semantic web, 9th international semantic web conference*. Shanghai.

Dey, K., & Abowd, G. D. (2000). Towards a better understanding of context and context-awareness. CHI 2000 workshop on the what, who, where, when, and how of context-awareness.

Fields, D. K., Kolb, M. A., & Bayern, S. (2001). *Web development with java server pages*. Manning Publications. ISBN: 193011012X.

Eberhart, A. (2003). Towards semantically enriched business logic. *Electronic Commerce Research and Applications, 2*, 288–301.

Ebrahimi, S., Villegas, N. M., Müller, H. A., & Thomo, A. (2012). SmarterDeals: A context-aware deal recommendation system based on the smarter context engine. CASCON '12IBM Corp., Riverton, NJ, USA, pp. 116–130.

Emmanouilidis, C., Koutsiamanis, R.-A., & Tasidou, A. (2012). Mobile guides: Taxonomy of architectures, context awareness, technologies and applications. *Journal of Network and Computer Applications*, 103–125.

Faraway, J. (2002). Practical regression and anova in R.

Forgy, C. (1991). Rete: A fast algorithm for the many pattern/many object pattern match problem. In P. G. Raeth (Ed.), *Expert Systems* (pp. 24–341). Los Alamitos, CA, USA: IEEE Computer Society Press.

Friedman-Hill, E. (2003). *Jess in action. Rule-based systems in java*. Manning Publications (pp. 32–33). Manning Publications. ISBN-10: 1930110898.

Furno, A., & Zimeo, E. (2014). Context-aware composition of semantic web services. *Mobile Networks and Applications, 19*(2), 235–248.

García-Crespo, Á., López-Cuadrado, J. L., Colomo-Palacios, R., GonzálezCarrasco, I., & Ruiz-Mezcua, B. (2011). Sem-Fit: A semantic based expert system to provide recommendations in the tourism domain. *Expert Systems with Applications, 38*.

Grosof, B. (2003). Representing E-commerce rules via situated courteous logic programs in RuleML. *Electronic Commerce Research and Applications*.

Guo, F., & Lu, Q. (2014). A novel contextual information recommendation model and its application in E-commerce. Customer Satisfaction Management.

Hand, A., Cardiff, J., Magee, P., & Doody, J. (2006). An architecture and development methodology for location-based services. *Electronic Commerce Research and Applications, 5*(3), 201–208.

Hawalah, A., & Fasli, M. (2014). Utilizing contextual ontological user profiles for personalized recommendations. *Expert Systems with Applications, 41*(10), 4777–4797.

Her, Y., Kim, S., & Jin, Y. (2010). A context-aware framework using ontology for smart phone platform. *International Journal of Digital Content Technology and its Applications, 4*(5).

Hirtle, D. (2006). TRANSLATOR: A TRANSlator from LAnguage TO Rules. In *Proceedings of the Canadian symposium on text analysis (CaSTA)*.

Horridge, M., & Bechhofer, S. (2009). The OWL API: A java API for working with OWL 2 ontologies. 6th OWLED.

Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet S., Grosof, B., & Dean, M. (2004). SWRL: A semantic web rule language combining OWL and RuleML. W3C member submission 21 May 2004, http://www.w3.org/Submission/SWRL/.

Hu, Y., Ching-Long, Y., Wolfgang, L. (2009). Challenges for rule systems on the web. In *Proceedings of the 2009 international symposium on rule interchange and applications, RuleML '09*.

Ilarri, S., Lllarramendi, A., Mena, E., & Sheth, A. (2011). Semantics in location-based services. *IEEE International Computing, 15*(6), 10–14.

Kaplan, B., & Maxwell, J. A. (2005). Qualitative research methods for evaluating computer information systems. *Evaluating the organizational impact of healthcare information systems* (pp. 30–55). New York: Springer.

Keßler, C., Raubal, M., & Wosniok, C. (2009). Semantic rules for context-aware geographical information retrieval. In P. Barnaghi et al. (Eds.). *European conference on smart sensing and context, EuroSSC 2009* (pp. 77–92). Springer. Volume 5741 of LNCS.

Kim, S., Suh, E., & Yoo, K. (2007). A study of context inference for web-based information systems. *Electronic Commerce Research and Applications, 6*, 146–158.

Kontopoulos, E., Bassiliades, N., & Antoniou, G. (2008). Deploying defeasible logic rule bases for the semantic web. *Data and Knowledge Engineering, 66*(1), 116–146.

Kwon, K., & Kim, C. (2012). How to design personalization in context of customer retention: Who personalizes what and to what extend? *Electronic Commerce Research and Applications, 11*, 101–116.

Kwon, O. B., Yoo, K. D., & Suh, E. H. (2006). ubiES: Applying ubiquitous computing technologies to an expert system for context-aware proactive services. *Electronic Commerce Research and Applications, 5*(3), 209–219.

Lassila, O. (2005). Applying semantic web in mobile and ubiquitous computing: Will policy-awareness help? Semantic web and policy workshop.

Lee, T., Lee, I., Lee, S., Lee, S., Kim, D., Chun, J., et al. (2005). *Electronic Commerce Research and Applications, 5*(1), 16–28.

Li, Y. M., & Jhang-Li, J. H. (2009). Pricing display ads and contextual ads competition, acquisition, and investment. *Electronic Commerce Research and Applications, 8*, 16–27.

Liagouris, J., Athanasiou, S., Efentakis, A., Pfennigschmidt, S., Pfoser, D., Tsigka, E., et al. (2011). Mobile task computing: Beyond location-based services and Ebooks. *W2GIS*, 124–141.

Liang, S., Fodor, P., Wan, H., Kifer, M. (2009). OpenRuleBench: An analysis of the performance of rule engines. WWW 2009 Madrid.

Liu, H., He, J., Wang, T., Song, W., & Du, X. (2012). Combining user preferences and user opinions for accurate recommendation. *Electronic Commerce Research and Applications*.

Liu, X., & Aberer, K. (2013, May). SoCo: A social network aided context-aware recommender system. In *Proceedings of the 22nd international conference on World Wide Web* (pp. 781–802). International World Wide Web conferences steering committee.

Liu, X., Liu, Y., Aberer, K., & Miao, C. (2013). Personalized point-of-interest recommendation by mining users' preference transition. In *Proceedings of the*

22nd ACM international conference on information & knowledge management (pp. 733–738). ACM.
Michael, K., & Michael, M. G. (2011). The social and behavioural implications of location based services. *Journal of Location Based Services, 5*(3–4), 121–137.
Motik, B., Grau, B. C., Horrocks, I., Wu, Z., Fokoue, A., & Lutz, C. (Eds.), OWL 2 web ontology language profiles, (2nd ed.), W3C Recommendation, 11 December 2012, http://www.w3.org/TR/owl2-profiles.
Myers, Michael D., & Avison, D. (1997). Qualitative research in information systems. *Management Information Systems Quarterly, 21*, 241–242.
Niforatos, E., Karapanos, E., & Sioutas, S. (2012). PLBSD: A platform for proactive location-based service discovery. *Location Based Services Journal, 6*.
Noguera, J. M., Barranco, M. J., Segura, R. J., Luis, M. (2012). A mobile 3D-GIS hybrid recommender system for tourism. TR-1.
Paschke, A., Kozlenkov, A., & Boley, H. (2007). 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.
Patkos, T., Bikakis, A., Antoniou, G., Plexousakis, D., & Papadopouli, M. (2007). A semantics-based framework for context-aware services: Lessons learned and challenges. *4th UIC-2007* (pp. 839–848). Springer. Vol. 4611 of LNCS.
Patton, E. W., & McGuinness, D. L. (2009). The mobile wine agent: pairing wine with the social semantic web. *2nd social data on the web workshop*. .
Serrano, D., Hervás, R., Bravo, J. (2011). Telemaco: Context-aware system for tourism guiding based on web 3.0 technology. Int. Workshop on Contextual Computing and Ambient Intelligence in Tourism.
Sherman, G. (2007). A critical analysis of XSLT technology for XML transformation. Senior Technical Report.
Skillen, K. L., Chen, L., Nugent, C., Donnelly, M., Burns, W., & Solheim, I. (2014). Ontological user modelling and semantic rule-based reasoning for personalisation of help-on-demand services in pervasive environments. *Future Generation Computer Systems*, 0167-739X, *34*, 97–109.
Tavakol, M., & Dennick, R. (2011). Making sense of Cronbach's alpha. *International Journal of Medical Education, 2*, 53–55.
Van Setten, M., Pokraev, S., & Koolwaaij, J. (2004). Context-aware recommendations in the mobile tourist application COMPASS. *Proc. of the 3rd international conference on adaptive hypermedia and adaptive web-based systems* (3137, pp. 235–244). Springer-Verlag. Lecture notes in computer science.
Viana, W., Filho, J. B., Gensel, J., Villanova-Oliver, M., & Martin, H. (2008). PhotoMap: From location and time to context-aware photo annotations. *Journal of Location Based Services, 2*(3), 211–235.
Viktoratos, I., Tsadiras, A., & Bassiliades, N. (2012). Personalizing location information through rule-based policies, rules on the web: Research and applications, RuleML 2012. *Proc. 6th international symposium on rules: Research based and industry focused* (pp. 215–223). Montpellier: Springer. LNCS 7438.
Woensel, V., Casteleyn, S., & Troyer, O. (2010). Applying semantic web technology in a mobile setting: "the person matcher". *Web Engineering*, 506–509.
Wu, S., Chang, C., Ho, S., & Chao, H. (2008). Rule based intelligent adaptation in mobile information systems. *Expert Systems with Applications, 34*, 1078–1092. Elsevier.
Yuan, Q., Cong, G., Ma, Z., Sun, A., & Thalmann, N. M. (2013). Time-aware point-of-interest recommendation. *Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval* (pp. 363–372). ACM.