Instituto Superior de Engenharia do Porto

Traceability-based Access Recommendation

Rafael Pinto Peixoto

Dissertação para obtenção do Grau de Mestre em Engenharia Informática, Área de Especialização em Tecnologias do Conhecimento e Decisão

Orientador: Eng.º Nuno Miguel Gomes Bettencourt
Co-Orientador: Doutor Nuno Alexandre Pinto da Silva

Júri:
Presidente:
Doutor João Paulo Jorge Pereira

Vogais:
Mestre Nuno Filipe Fonseca Vasconcelos Escudeiro
Eng.º Nuno Miguel Gomes Bettencourt
Doutor Nuno Alexandre Pinto da Silva

Porto, Outubro de 2012

Aos meus pais e irmão ...

Resumo Alargado

Devido à grande quantidade de dados disponíveis na Internet, um dos maiores desafios no mundo virtual é recomendar informação aos seus utilizadores. Por outro lado, esta grande quantidade de dados pode ser útil para melhorar recomendações se for anotada e interligada por dados de proveniência.

Neste trabalho é abordada a temática de recomendação de (alteração de) premissões acesso sobre recursos ao seu proprietário, ao invés da recomendação do próprio recurso a um potencial consumidor/leitor. Para permitir a recomendação de acessos a um determinado recurso, independentemente do domínio onde o mesmo se encontra alojado, é essencial a utilização de sistemas de controlo de acessos distribuídos, mecanismos de rastreamento de recursos e recomendação independentes do domínio.

Assim sendo, o principal objectivo desta tese é utilizar informação de rastreamento de ações realizadas sobre recursos (*i.e.* informação que relaciona recursos e utilizadores através da Web independentemente do domínio de rede) e utilizá-la para permitir a recomendação de privilégios de acesso a esses recursos por outros utilizadores. Ao longo do desenvolvimento da tese resultaram as seguintes contribuições:

- A análise do estado da arte de recomendação e de sistemas de recomendação potencialmente utilizáveis na recomendação de privilégios (secção 2.3);
- A análise do estado da arte de mecanismos de rastreamento e proveniência de informação (secção 2.2);
- A proposta de um sistema de recomendação de privilégios de acesso independente do domínio e a sua integração no sistema de controlo de acessos proposto anteriormente (secção 3.1);
- Levantamento, análise e especificação da informação relativa a privilégios de acesso, para ser utilizada no sistema de recomendação (secção 2.1);
- A especificação da informação resultante do rastreamento de ações para ser utilizada na recomendação de privilégios de acesso (secção 4.1.1);
- A especificação da informação de *feedback* resultante do sistema de recomendação de acessos e sua reutilização no sistema de recomendação(secção 4.1.3);
- A especificação, implementação e integração do sistema de recomendação de privilégios de acesso na plataforma já existente (secção 4.2 e secção 4.3);
- Realização de experiências de avaliação ao sistema de recomendação de privilégios, bem como a análise dos resultados obtidos (secção 5).

Abstract

Due to the large amount of available data in the internet, one of the biggest challenges in the virtual world is to recommend information to the user. On the other hand this large amount of data can be useful to improve recommendations if it is semantically described and inter-related. To describe and relate this information, provenance information is fundamental.

Several resources are not totally recommendable but can be recommended a specific type of access to them. So the cross-domain information provenance, cross-domain access control and cross-domain access recommendation are leading keys to improve cross-domain recommendation.

The main goal of this thesis work is to use automatic traceability information of actions that are performed over resources in order to relate users and resources over the Web without relying on the domain and use this information to recommend access privileges to other users.

Keywords: Traceability, Access Policy Recommendation, Access control

Acknowledgements

In overall I would like to thank all the people that in some way gave me their support and contribution during my entire work at GECAD (Knowledge Engineering and Decision Support Group).

First, I would like to thank both my advisors for all their support and advice. To Nuno Bettencourt, especially for his support during the elaboration of this work and to Nuno Silva, especially for all his effort and time spent advising and constructively criticizing this work.

Second, I would like to thank the opportunity of working in the the projects OOBIAN - Living Knowledge (QREN 12677) and AAL4ALL - Padrão de Cuidados Primários para serviços de Ambient Assisted Living (QREN 13852) and their support to make this work possible.

A special thanks to my mother Luzia Pinto, to my father Aventino Peixoto and brother Carlos Peixoto, for their unconditional support. Last but not least, I would like to thank all my friends for all the good moments away from work.

Contents

List of Figures

List of Tables

Glossary

ABAC Attribute-Based Access Control. 7, 60, 67–70

API Application Programming Interface. 21–23

CSV Comma-separated values. 24

DAC Discretionary Access Control. 65, 67

DL Description Logic. 61, 69

DSoD Dynamic Separation Of Duty. 67, 69, 70

FOAF Friend of a Friend. 2, 5, 6, 8, 11, 29, 32, 38

HTTP HyperText Transfer Protocol. 2

HTTPS HyperText Transfer Protocol Secure. 6, 42

JSON JavaScript Object Notation. 24

LOD Linked Open Data. 6–8, 10, 11, 28

MAC Mandatory Access Control. 65

OWL Web Ontology Language. 69, 71

PAP Policy Administration Point. 6, 7, 27, 28, 31, 32, 35, 42

PDP Policy Decision Point. 6, 7, 41

PEP Policy Enforcement Point. 6–8, 43, 68

PIP Policy Information Point. 6–8, 31, 32, 35, 39, 41–43

PKI Public-Key Infrastructure. 6

PROV-O PROV Ontology. 37

PRP Policy Recommendation Point. 27–29, 31, 32, 34–36, 38–40, 42, 43, 45, 52, 53, 59, 79

RBAC Role-Based Access Control. 7, 65, 67–71

RDF Resource Description Framework. 11, 21, 22, 24

REST Representational State Transfer. 23

ROWLBAC Role Based Access Control in OWL. 38, 69–71

SoD Separation of Duty. 67

SSL Secure Sockets Layer. 5, 6

SSoD Static Separation Of Duty. 67, 69, 70

URI Uniform Resource Identifier. 10, 11, 29

WOT Web of Trust. 14

XML Extensible Markup Language. 24

Chapter 1

Introduction

With current information systems, such as the Web [Berners-Lee and Cailliau, 1990], more and more information is stored on a large scale. The unstructured data and pages of millions of authors about hundreds of subjects/topics in the web is increasing. This makes that information systems spend a long time on resource intensive tasks to search and retrieve information on the Web. Due to the wide range of topics in the Web, users often do not understand, much less are comfortable to choose among various suggested available alternatives that are presented by search engines. To reduce the doubts and needs when trying to choose among various alternatives, the user typically relies on suggestions given by others, which can be received directly [Shardanand and Maes, 1995] or by recommendation texts, opinions of reviewers, books and newspapers, among others. Over the years, recommender systems have tried to address how to proceed in these cases. Recommender systems have tried to address these problems by increasing the capacity and effectiveness of this recommendation process and exploiting the social relationships between humans [Resnick and Varian, 1997].

In a typical recommender system, users provide items as input, the system gathers and processes it to individuals considered as potentially interested in such information. One of the great challenges of this type of system is to perform the appropriate mix between the users expectation and the products, services and persons to be recommended to them, *i.e.* the definition of the relationship of interest is one of the problems to be dealt with in a Recommender System.

Currently, recommender systems are widely used in electronic commerce [Adomavicius et al., 2011][Linden et al., 2003][Schafer et al., 2001], using different techniques to find the most suitable products for their customers and thus increase profit. Introduced in July 1996, "My Yahoo" was (one of) the first website to use a recommender system in large proportions using the personalization strategy [Manber et al., 2000]. Presently,

a large number of websites use recommender systems to make suggestions to different kinds of users. There are other areas in which recommendation is relevant such as the access recommendation to resources (i.e. anything that is identifiable, such as webpages, documents, fact) [Bettencourt and Silva, 2010]. The access recommendation to resources is the process of recommending (changes to) access privileges to resources to the owner, instead of recommending the resource itself to the consumer/reader.

In [Bettencourt and Silva, 2010] it is proposed a framework that:

- provides cross-network domain [1] identification and authentication;
- provides cross-domain access control to web resources;
- captures the users actions (*e.g.* upload, read, update) upon the web resources, giving rise to traceability information.

To provide these features in a cross-domain distributed way, traditional authentication and access control systems included in web applications are not sufficient as they rely on centralized and proprietary databases for authentication (*i.e.* HyperText Transfer Protocol (HTTP) Authentication [Franks et al., 1999]) and authorization. To reduce such problem, the FOAF+SSL protocol [Story et al., 2009] was suggested, providing a cross-domain and (globally) decentralised authentication protocol, built upon the usage of Friend of a Friend (FOAF) profiles [Brickley and Miller, 2010]. This also reduces multiple accounting on web applications as it provides a single user identity across different web domains.

While FOAF+SSL provides a decentralized authentication protocol, it is not enough to ensure or verify if a person has access to a particular web resource. Nowadays a user can only list Web resources submitted for a given web domain through mechanisms provided by each web data *island*, making it impossible to list all the user resources (*i.e.* all the resources s/he owns regardless of the domain where they are hosted). To list all resources made available by a user, it is necessary to relate the users and all their resources. One way to relate users and resources is through the actions performed by users over those resources. For that, an automatic traceability acquisition framework is necessary as described in [Bettencourt et al., 2012]. Such framework uses action sensors to intercept the user actions over resources in order to create traceability annotations. These annotations provide more information about a user and relate users to their resources using a unique user identity. It is our conviction that traceability information can improve the recommendation process of access privileges?

[1] From now the expression "cross-network domain" will be substituted by "cross-domain" as it sufficiently (and better) captures the intended meaning.

This work aims to enhance the previously described framework with features capable of predicting access privileges for resources and recommend those access privileges to other users (*i.e.* owners). The recommendation system shall notify the owner of a resource that there is a user that could benefit from accessing a specific resource. The interested user may be related to the resource owner (*i.e.* Friend), or may not have any relationship with the resource owner.

1.1 Contributions

The work described in this thesis has the following contributions:

- State of the art about recommendation and recommendation systems of potential usage for resource access privilege recommending systems;
- State of the art about resource traceability and provenance information systems;
- Proposal of a domain independent resource access privilege recommender system and its deployment on the access control system previously proposed;
- Retrieval, analysis and specification of information related with access control privileges to be used in the recommender system;
- Specification of the traceability information that will be used to recommend access privileges;
- Specification of feedback information from the access recommender system and the re-utilization of such information in the recommender system;
- Specification, implementation and integration of the access privileges recommender system in the previous access control and management framework;
- Access control recommendation system evaluation and the resulting data analysis.

1.2 Thesis Overview

This thesis is organised in six chapters. Chapter one is the introduction where is presented the main objectives, contributions and gives a brief description of this work. Chapter two presents an overview of the access control framework used in this work, the concept of traceability and its acquisition framework. Also, a systematization of concepts and brief state of the art in recommender systems are presented. Chapter three describes the proposed architecture for recommending access privileges to resources based

on traceability information and its integration with the original framework. Chapter four describes the instantiation of the proposed architecture and the adopted data model and respective management process. Chapter five describes the experiments carried out to evaluate the proposed architecture and the performed analysis. Finally, Chapter six presents the conclusions and future work.

Chapter 2

Technological Background

This chapter addresses three subjects in three sections:

- **Access Control**: Presents an overview of the access control concepts and more details about the framework used in this work to ensure access control over the web resources;
- **Traceability**: Presents the concept of provenance and traceability and addresses its representation in a conceptual way;
- **Recommendation**: Presents a brief state of the art on recommender systems, techniques, strategies, application in e-commerce and tools.

2.1 Access Control

Access control is about regulating which resources may be accessed and by whom. This implies Authentication and Authorization.

In [Bettencourt and Silva, 2010] the authors have proposed a decentralized framework capable of providing cross-domain authentication, authorization and access control management.

This framework relies on FOAF+SSL authentication. FOAF + SSL [Story et al., 2009] is a secure authentication protocol that allows the construction of a distributed social network, open and secure, based in the Web of Trust. While FOAF [Brickley and Miller, 2010] relates to user profiles description, Secure Sockets Layer (SSL) [Dierks and Allen, 1999] is a widely used protocol on the Internet to provide secure communications between

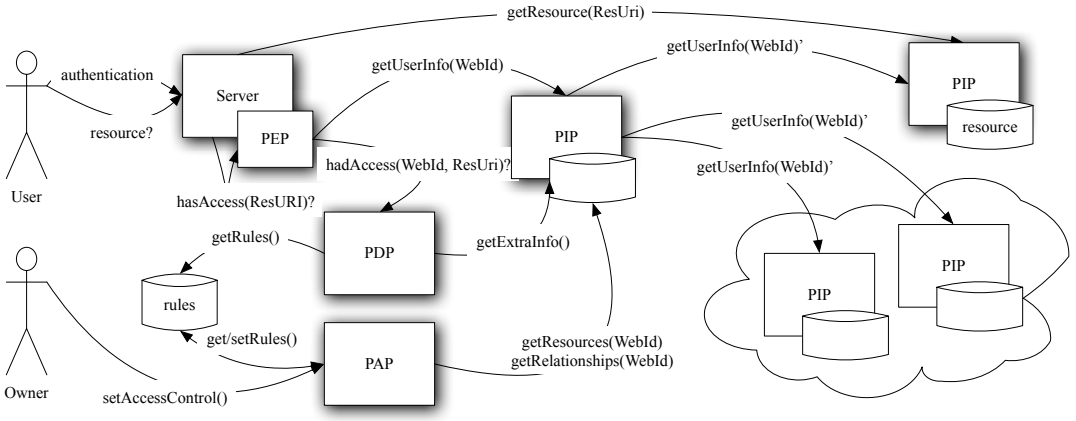

FIGURE 2.1: Access control framework overview

clients and servers. SSL Client Authentication consists in authenticating users by checking the contents of their client certificates. A typical client certificate contains detailed identification information about the user, the organization that issued the certificate, and a public key. FOAF+SSL uses FOAF profiles to create a trust network in order to substitute the traditional role of a Certificate Authority (CA) in typical Public-Key Infrastructure (PKI) [Dierks and Allen, 1999; Housley et al., 2002] schemes. FOAF+SSL requires a SSL client certificate and extends it in order to include a reference to a FOAF profile. The FOAF+SSL protocol can easily be used over HyperText Transfer Protocol Secure (HTTPS) protocol [Rescorla, 2000]. In addition, as the authentication is decentralised it reduces multiple accounting on web applications by providing a single user identity across different web domains.

This framework has four main components: (i) Policy Information Point (PIP), (ii) Policy Administration Point (PAP), (iii) Policy Enforcement Point (PEP) and (iv) Policy Decision Point (PDP), as depicted in Fig. 2.1.

The next four subsections present an overview and some analysis on the four main components of the access control framework proposed in [Bettencourt and Silva, 2010].

2.1.1 Policy Information Point

A PIP component is responsible for retrieving information that is not available inside a local system like Linked Open Data (LOD) repositories, websites, databases or any other kind of information repository that may be registered on the component. This component acts as a broker interface to existing repositories that have information about resources, authors, web servers, etc. PIP components not only act like data providers

brokers, as they are also capable of creating data and making it available in those information repositories. PIPs are also responsible for retrieving, creating and announcing new provenance information about any resource and replicating it over to other information repositories. These components also provide querying capabilities to existing information repositories, therefore being able to answer to questions like return a list of all resources for a given user or retrieve a resource's author.

2.1.2 Policy Administration Point

The PAP will grant the user with services to manage access privileges over existing resources. The main service is to create, modify or remove privileges to resources. The access privilege creation or modification has to prevent possible inconsistencies caused by new privileges on the access control system and creating, modifying or removing existing access privileges.

2.1.3 Policy Enforcement Point

The PEP component is responsible for tasks such as authentication, authorization and creating resource provenance information. Each PEP provides means of validating FOAF+SSL authentication for every user, even if the user is not registered on the domain proprietary registration application. For each request a server receives over resources (*i.e.* downloading, uploading) a PEP is responsible for intercepting it. This component uses action sensors to intercept users actions over resources and allows resource traceability information generation. This provenance information is registered by PIP on LOD repositories.

2.1.4 Policy Decision Point

The PDP is responsible for making the decision of granting or not access to a resource. According to PAP access privileges and the user supplied by the PEP component, the PDP evaluates whether or not the user should have access to a resource. If more information is needed to evaluate access, the PDP may contact any PIP point to obtain extra information. PDPs' implicit authorization sub component can use different access control systems such as Role-Based Access Control (RBAC) or Attribute-Based Access Control (ABAC) or their combination [Klenk et al., 2009]. Despite these models are not used in this work.

2.2 Traceability and Provenance information

Provenance is defined as the source of something [Webster, 2012]. The traceability information can be used to define the provenance (information) about something (*e.g.* a web resource or a software feature). For example, software requirements traceability refers to the ability to describe and follow the life of an artefact, in both forward and backward directions [Gotel and Finkelstein, 1994].

Accordingly, there are two main concerns to address here: (i) Traceability capturing and (ii) the Traceability (conceptual) model.

2.2.1 Traceability Capturing

To capture and publish traceability information is proposed in [Bettencourt et al., 2012] a framework that can capture and publish traceability information from the user's actions upon resources. This framework uses: (i) FOAF profiles for describing each user and his/her relationships with other users, (ii) FOAF+SSL to provide single user and cross-domain identity and (iii) Linked Open Data repositories to store generated traceability information. To capture traceability information, this framework makes use of action sensors (responsible for intersecting user actions on resources) and metadata generators for each resource. Such framework relies on a PIP component, capable of listening publishing requests from the PEP action sensors, retrieve and coalesce the data from different LOD and respond accordingly. Such provenance acquisition framework should be deployed on (i) the web servers where the web applications are installed by reusing the PEP and (ii) on the webservers where the PIP component is deployed. The deployment of this traceability acquisition framework is presented in Fig. 2.2.

2.2.2 Traceability Information

In [Moreau and Missier, 2012] the authors propose a conceptual data model to represent entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability or trustworthiness. To represent the provenance data model [Moreau and Missier, 2012] proposed an ontological[1] model in [Lebo et al., 2012] named PROV-O. This ontology provides a set of classes, properties, and restrictions that can be used to represent and interchange provenance information generated in different systems and under different contexts. It can also be specialized to

[1]Ontologies are "a formal, explicit specification of a shared conceptualization". More information about ontologies can be found in Appendix 1

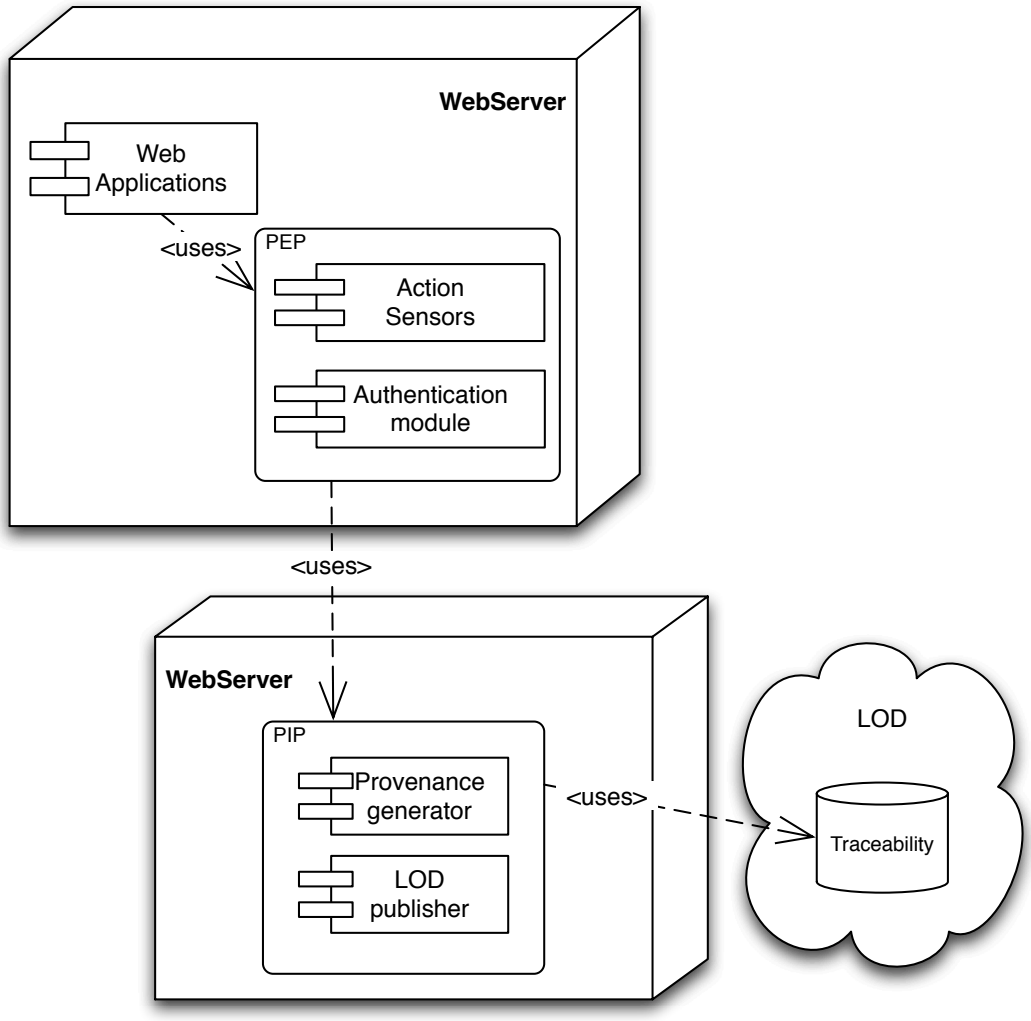

FIGURE 2.2: Traceability acquisition framework deployment

create new classes and properties to model provenance information for different applications and domains. The Provenance Vocabulary core ontology presented by [Hartig and Zhao, 2012] is designed as a Web data specific specialization of the PROV-O ontology. The Provenance Vocabulary core ontology classes and properties are domain specific extensions of the more general concepts introduced in PROV-O ontology. Fig. 2.3 presents an overview of the Provenance core ontology domain.

This information can be useful to recommend access privileges because they rely on activities previously performed over resources that can be used to predict user interests and behaviours.

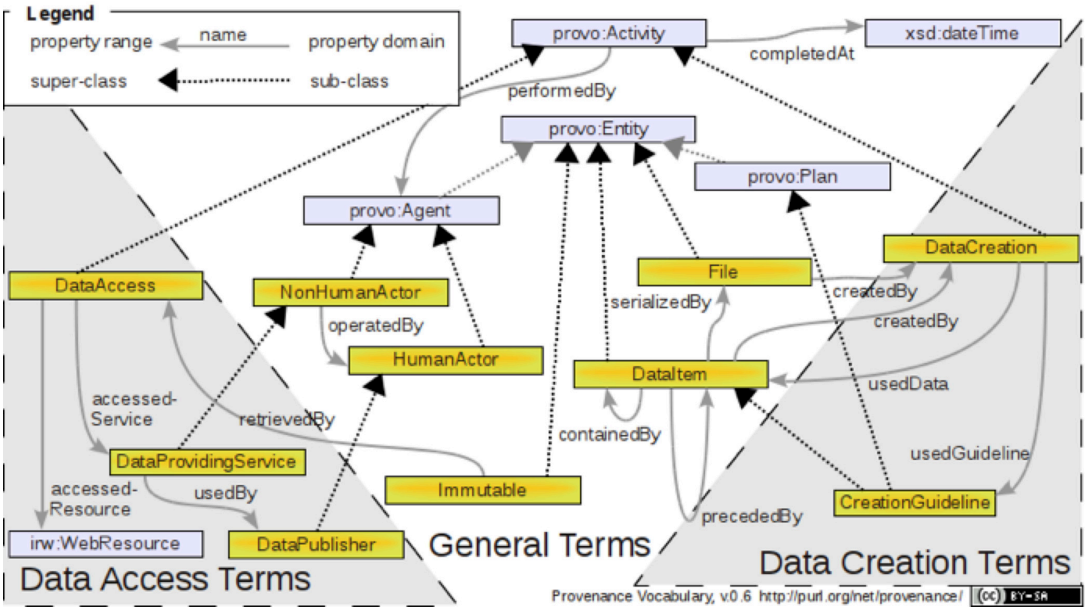

FIGURE 2.3: Provenance Core Ontology

2.3 Recommendation

Information overload becomes increasingly severe nowadays because of mass media and global communications facilities, which exceed the human ability to filter and dissect relevant information from the irrelevant one. Consequently, significant research [Blair, 1979] efforts have been done to build automated filtering systems that provide only relevant information to users. In the last years, recommender systems helped increasing the capacity and effectiveness in the resource nomination process [Resnick and Varian, 1997]. Modern recommender systems are Web-based but mostly do not use data from different domains. Cross-domain recommender systems must consider user and item models in order to mediate the user data through different systems and application domains [Berkovsky et al., 2006][Berkovsky et al., 2007][Berkovsky et al., 2008]. Using this information, a recommender system can produce recommendations from user preferences collected in one domain to generate recommendations in a different one. Based on the information available on cross-domain LOD repositories, [Fernández-Tobías et al., 2011] present an on-going research work about a generic knowledge-based description framework upon semantic networks to provide cross-domain recommendations.

In the next sub-sections it will be described:(i) the information that can be used in recommender systems, (ii) the recommendation techniques that will compute recommendations, (iii) the recommendation strategies to show users those recommendations, (iv) the recommender system measures used to evaluate recommender systems and (v) some recommendation systems.

2.3.1 Recommendation Information

Recommender systems gather various kinds of input information in order to build their recommendations. Such information is primarily about the resources to recommend and the users who will receive these recommendations.

A resource, in this case a Web Resource, can be globally identified by an Uniform Resource Identifier (URI). An URI [Berners-Lee et al., 2005] is a compact sequence of characters that identifies an abstract or physical resource. Resources and users can be related by using Resource Description Framework (RDF) [Needleman, 2001] triples (*i.e.* subject-predicate-object), where subject and predicate are URI, and object can be either a URI or a value. To store these triples, LOD [Bizer et al., 2009] repositories can be used.

2.3.2 Recommendation Techniques

Recommendation systems have been applied over the past decade in many areas using many techniques to generate the recommendations. Some of these techniques have evolved from existing techniques in information retrieval systems whose main objective is to retrieve useful or relevant information to the user [Adomavicius et al., 2011]. Common recommendation techniques are based on filtering such as Content-Based Filtering, Collaborative Filtering or Hybrid filtering. Others such as Knowledge Discovery, Bayesian Belief Networks, Context, Trust or Social Networks can also be used in recommender systems. A brief description of these techniques is presented in the next sub-sections.

2.3.2.1 Content-Based Filtering

Content-Based Filtering technique [Herlocker et al., 2000] generate descriptions of the resource contents and compare these descriptions with the users interests in order to verify the resource relevancy. Content-based filtering can use ontologies for the representation of users and resources in order to represent the classification according to their relevance in the field [Shoval et al., 2008]. The filtering method proposed by the authors in [Shoval et al., 2008] considers the hierarchical distance, or proximity between the concepts of user profile and concepts in the resource profile and uses a hierarchical ontology.

2.3.2.2 Collaborative Filtering

Unlike Content-based Filtering, Collaborative Filtering does not exactly require an understanding or recognition of the content of resources [Herlocker et al., 2000] because the essence of collaborative filtering systems is the exchange of experiences among people who have common interests and not on the content of resources. In [Sieg et al., 2010], the recommendation is based on a collaboration pattern, where the user similarities are calculated based on their scores between ontology concepts. The experimental recommendation results of [Sieg et al., 2010] show a significant improvement in recommendation accuracy compared to standard collaborative filtering.

2.3.2.3 Hybrid Filtering

Hybrid Filtering techniques combine the strengths of the various filtering approaches with the aim of creating a system that can better meet the needs of the user. In [Burke, 2002] the author presents seven types of different hybrid approaches to hybrid recommendation and in [Airoldi et al., 2011] the authors present another two different approaches for building hybrid collaborative plus content recommender systems. The purpose of hybrid filtering approaches is to produce relevant recommendations, while overcoming the "Cold-Start" new resource issue. Cold-start issue is the lack of initial information about users and items, common in the collaborative recommender systems, that leads to low precision recommendations [Schein et al., 2002].

2.3.2.4 Knowledge Discovery

A Knowledge Discovery System is a system that builds knowledge that was not implicit or explicit in their algorithms in the current representation of domain knowledge. When working with web recommender systems, discovery of knowledge is an important resource for the discovery of relationships between resources, users and between users and resources. Through the mining of log files [Yang and Parthasarathy, 2003], for example, we can get detailed knowledge about the users who have logged on to a website. This knowledge can be used for product offerings customization [Rucker and Polanco, 1997], web sites structuring according to the Internet user profile and by customizing pages content.

2.3.2.5 Bayesian Belief Networks

Bayesian networks [Pearl, 1985] are powerful tools for modelling causes and effects in a variety of fields. Compact networks are likely to capture the probabilistic relationship between variables, as well as historical information about their relationship. Bayesian networks are very effective to model situations where some information is already known and the input data are uncertain or partially unavailable. These networks provide a consistent semantics for representing causes and effects through an intuitive graphical representation. Due to all these features, the Bayesian networks are increasingly used for a wide variety of domains where inference is necessary. An important fact of Bayesian networks is that they are not dependent on accurate historical information or current evidence. In other words, Bayesian networks can often produce very convincing results even when the historical information in the tables of conditional probability or evidence is not known exactly.

2.3.2.6 Context-Based Recommendation

Contextual information is useful in information retrieval [Jones, 2005], although the decisions taken in most information retrieval systems are based only in consultation and collection of documents, whereas information about the context of research is often ignored [Akrivas et al., 2002]. In web search, the context is considered as a set of topics potentially related to the search term [Maamar et al., 2006]. In order to provide recommendation based on semantic context-dependent content some recommendation approaches like [Yu et al., 2007] [Loizou and Dasmahapatra, 2006] use ontologies. In [Loizou and Dasmahapatra, 2006] the authors propose an ontology-based system that contains contextual information about the recommendation process and items. The contextual information processed through heuristic rules applied to vector spaces, allow the system to dynamically place a given recommendation.

2.3.2.7 Trust-Based Recommendation

According to the author in [Sinha and Swearingen, 1999] people rely more on recommendations from people they trust (friends) than in an online recommendation system based on generating recommendations from anonymous people with similar characteristics. In [Bedi et al., 2007] Bendi et al. proposes a trust-based recommender system for the semantic web, based on ontologies and using the Web of Trust (WOT) [Guha et al., 2004] to generate recommendations. In contrast to current trust models that ignore

user feedback [Bedi et al., 2007], in [Moghaddam et al., 2009] the authors propose a two-dimensional trust model that dynamically gets updated based on user's feedback.

2.3.2.8 Social Recommendation

Integrating social networks with recommendation systems can improve the performance of recommendation systems. The accuracy of the recommendation process can be improved by the user's relations depth obtained from social networks, thereby improving the understanding of the user behaviour [He and Chu, 2010]. As proof of concept, in [Fazel-Zarandi et al., 2011] the authors have developed a prototype of a recommender grounded in social science. In the systems presented in [Fazel-Zarandi et al., 2011] [Noor and Martinez, 2009], the Semantic Web and ontologies can help with the representation of context and interpretation of social data. In this case it is possible to avoid the "Cold-Start" problem.

2.3.3 Recommendation Strategies on E-commerce

Considering the wide application, acceptance and validity of recommender systems in scope of e-commerce [Schafer et al., 2001], this section aims to survey and analyse the adopted recommendation strategies in order to devise valid and meaningful recommendation strategies for access privilege recommendation based on traceability.

The main objectives of recommendation systems on e-commerce are fidelity and the consequent increase of company profits. Different strategies can be used to provide information to a user, each requiring a different degree of complexity in the treatment of information collected.

2.3.3.1 Recommendation list

This strategy maintains resource lists organized by type of interest. In this case, it is not required a deep analysis of user data to create these lists, only the observation of the most popular types of resources, and sort these into groups such as "best-selling," "gift ideas", "most viewed" among others. Fig. 2.4 presents the Amazon.com website without authentication performed. In this situation, a recommendation list (marked with a red rectangle) is presented to the user.

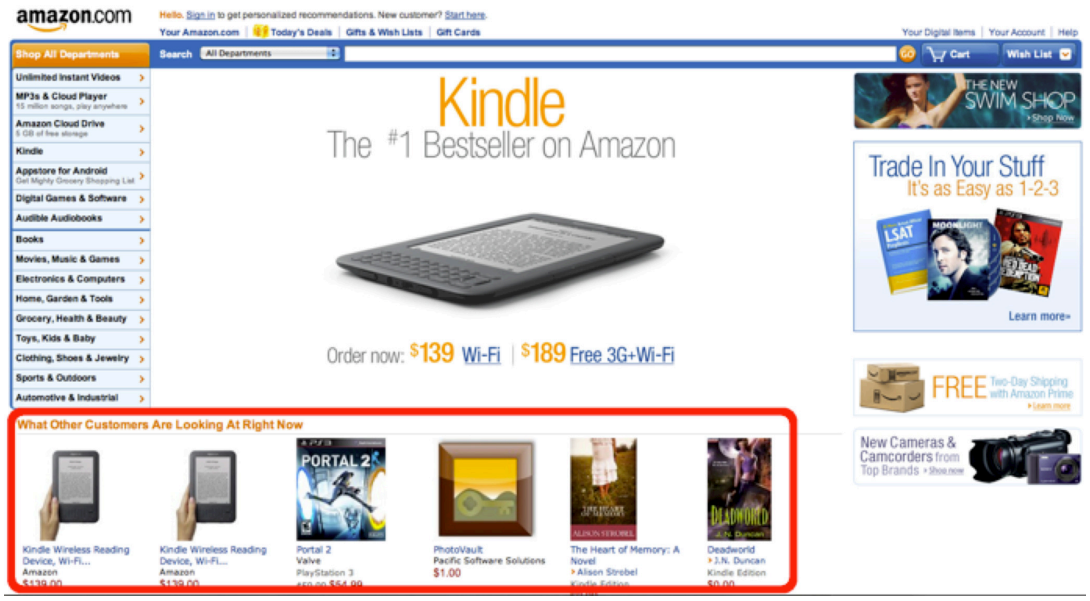

FIGURE 2.4: Amazon.com recommendation lists

FIGURE 2.5: Recommendation based on user evaluation of Amazon.com

2.3.3.2 User Reviews

User review is one of the best strategies used in recommendation systems, which in addition to buying a product, the user also makes a comment about the item purchased. Customer reviews are very useful for other users to ensure the quality and usefulness of the products sold. However, for a system to work correctly based on user comments, it is necessary to verify the provided opinions veracity. Fig. 2.5 presents the evaluation of system users based on stars and bar graph from Amazon.com website.

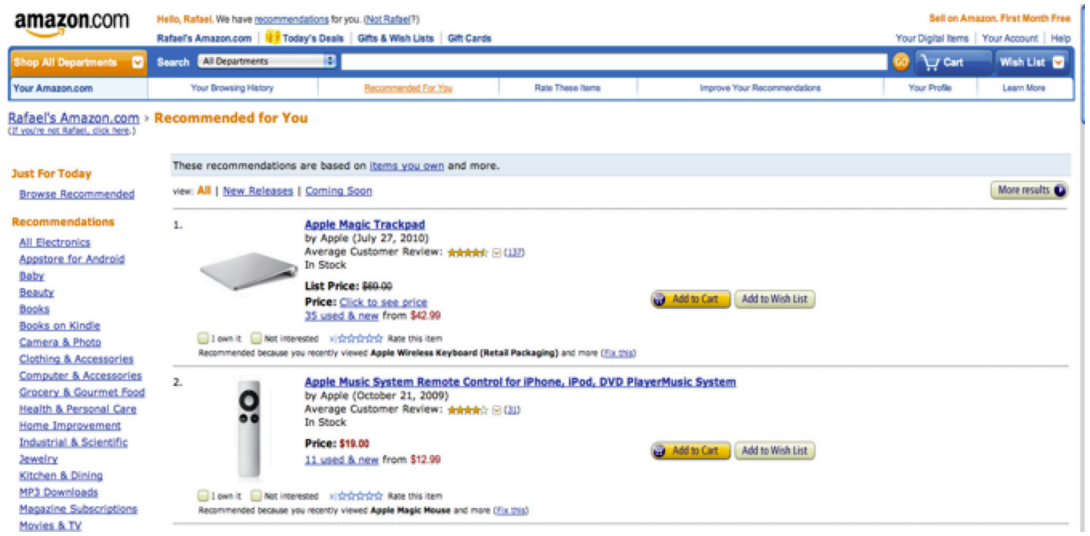

FIGURE 2.6: Recommendation based on user actions of Amazon.com

2.3.3.3 Recommendation based on user actions

This type of recommendation is offered in a section (*i.e.* web page section) devoted entirely to suggestions made specifically for the user. Two types of recommendations are possible in these sections: those made from implicit or explicit preferences. Fig. 2.6 shows the example of a recommendation system "Recommended For You" from amazon.com where users are brought suggestions from data obtained implicitly. It also presents the user with a justification for the given recommendation, and if the user does not agree with the recommendation s/he can remove that recommendation.

2.3.3.4 Content Association

Recommendations based on content association are present, for example, when a person buys a computer and the system recommends products based on products that other users purchased with the product being viewed. In the case of Fig. 2.7, users who bought the laptop also tend to buy a mouse, which makes sense and should be presented to the user.

2.3.4 Recommender Systems measures

One undeniably effective way for the evaluation of recommendation systems is through a process of real usage. Unfortunately, it is very difficult and/or expensive to carry out such tests, because both the quantity and the correct selection of individuals for the creation of a significant sample are quite difficult to reach. Still, it is important to obtain

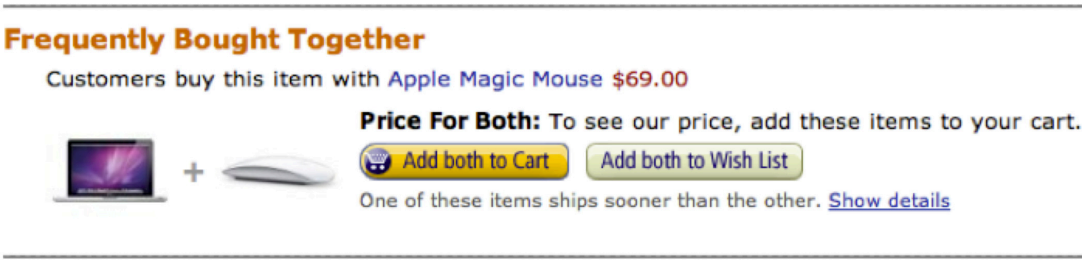

FIGURE 2.7: Recommendation based on content association of Amazon.com

TABLE 2.1: Confusion Matrix

	Recommended	Not Recommended
Used	True-Positive (TP)	False-Negative (FN)
Not Used	False-Positive (FP)	True-Negative (TN)

measures about the system performance before it is used, in order to estimate if the recommendations generated are appropriate. This led to the emergence of approaches for testing recommender systems through simulations, allowing estimating the systems behaviour. The most common measures are presented in the subsections below.

2.3.4.1 Prediction Accuracy

The majority of recommender systems are based on prediction engines that predict user interest on resources. Prediction accuracy is one of the most discussed properties in recommender systems because the main goal is to maximize predictions accuracy. In the case of recommending interesting resources to users the accuracy can be measured with resource to a Confusion Matrix like the presented in Table 2.1.

In Table 2.1 the True-Positive result is the number of recommended items that are useful, the False-Positive is the number of recommended items that will not be useful, the False-Negative are the number of useful items that will not be recommended and the True-Negative is the number of useful items that will not be recommended. With this data it is possible to calculate precision and recall of Recommender System.

$$Precision = \frac{TP}{TP + FP} \tag{2.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.2}$$

2.3.4.2 Coverage

Coverage is the result proportion that is likely to be recommended in relation to the set of all known resources. The coverage can be measured by: (i) the Item Space Coverage and by (ii) User Space Coverage.

Item Space Coverage , the most popular formula for measuring inequality in proposals is the Gini Index presented by the Formula 2.3 where p(i) is the probability of appearing as recommendation and items ij are ordered according to the increasing p(i).

$$G = \frac{1}{N-1}\sum_{j=1}^{n}(2j - n - 1)p(i_j) \tag{2.3}$$

The Shannon Entropy, presented by Equation 2.4 is also used as a measure of disproportion. H is equal to 0 when the single item is always chosen and log n if probabilities are distributed equally.

$$H = -\sum_{i=1}^{n} p(i)logp(i) \tag{2.4}$$

The User Space Coverage is the proportion of users or user interactions for which the system can recommend items. This coverage can be measured by the richness of the user profile required to make a recommendation.

2.3.4.3 Support

Support defines how often a set of items <x,y> appear together in different item-sets such as baskets [Agrawal et al., 1993](*e.g.* user actions).

2.3.4.4 Confidence

Confidence is the trust measure that the recommendation system has in its predictions. The lower confidence in a recommended resource can lead the user to research about the resource before accepting the recommendation. The simplest measure of catalogue confidence is the percentage of all items that can ever be recommended.

The confidence in a recommendation rule that relates x and y can be obtained by Equation 2.5.

$$Confidence(x \rightarrow y) = \frac{Support(< x, y >)}{Support(x)} \tag{2.5}$$

2.3.4.5 Trust

Trust is the user trust in the recommendation system. This trust can be acquired with the recommendation of multiple reasonable recommendations. This can be very important to future recommendations.

2.3.4.6 Novelty

Novel recommendations are recommendations of resources unknown to the user [Konstan et al., 2006]. Novel resources may be beneficial to users. However, a resource may be new to the user, but still worthless. One approach would be to consider novelty only among the relevant resources [Zhang et al., 2002]. In [Vargas and Castells, 2011] are presented measures to measure novelty in recommendation. One of this measures is presented in Equation 2.6 that is a measure of overall recommendation novelty where R is the list of recommended items, i is an item of that list and the function p(i) is the probability that i is used.

$$novelty(R) = -\sum_{i \in R} p(i|R) log_2 p(i) \tag{2.6}$$

2.3.4.7 Serendipity

Serendipity is a measure of how surprising the recommendations are. In other words can be the amount of relevant information that is new to the user in a recommendation. In [Murakami et al., 2008] the authors propose a new measure to measure serendipity of recommender lists called *unexpectedness*, which is the distance between the results produced by the method to be evaluated and those produced by a primitive prediction method. To calculate *unexpectedness*, some notions must be defined:

si(i = ··· N) denote the *i-th* ranked item in the recommendation list;

Pr(si) denote *si's* belief generated by the prediction method;

Prim(si) denote *si's* belief generated by the primitive prediction method.

Here, belief means the degree to which the recommender system confidently recommends each item. The relation to the user's preferences is $isrel(si) \in [0, 1]$, where *isrel(si) = 1* means that *si* is related to the user's preferences and *isrel(si) = 0* means that it is not. Unexpectedness of each item is defined as presented in Equation 2.7.

$$unexpectedness = \frac{1}{N} \sum_{i=1}^{n} max(Pr(s_i) - Prim(s_i), 0).isrel(s_i) \tag{2.7}$$

2.3.4.8 Diversity

Diversity is the recommendation of resources that are different between them. In some cases suggesting a set of similar resources may not be as useful for the user, because it may take longer to explore the range of items. The recommendation of diverse resources can be beneficial but depending on what the system wants to recommend. Diversity can be measured in different ways. In [Vargas and Castells, 2011] to measure diversity based on Novelty are proposed but in [Kowalczyk and Schut, 2011] the authors present a process to measure diversity under different users choices in real rating data.

2.3.4.9 Utility

The utility of a recommendation is the value that the system or the user gains from a recommendation. It can be measured clearly from the perspective of the recommendation engine or the recommender system owner.

2.3.4.10 Robustness

Robustness is the recommendation stability in the presence of false information [OMahony et al., 2004]. This information is typically inserted with the purpose to influence the recommendations and manipulate the recommender system.

2.3.4.11 Privacy

User preferences must stay private in order to avoid that external applications can use the recommendation system to learn something about the preferences of a specific user.

2.3.4.12 Adaptability

A recommender system must operate in a rapidly changing environment where trends in interest over items may shift. An example is the news recommendation. Adaptability is the system ability to adapt to an evolving environment of this type.

2.3.4.13 Scalability

One of the main functions of a recommender system is to help users in a large environment with several resources. The scalability in a recommender system is the ability to adapt to large scale of items to recommend.

2.3.5 Recommendation libraries

Recommendation libraries provide functions and algorithms to build recommender systems. The next sub-sections present some recommendation libraries.

2.3.5.1 Apache Mahout

Apache Mahout[1] is a Java library that has four main use cases: Recommendation, Clustering, Classification and Frequent item set mining. Recommendations can be done by collaborative or content data loaded from a database or formatted files.

2.3.5.2 GUSTO

GUSTO[2] is a set of Application Programming Interface (API)s to build intelligent web applications. Gusto uses a semantic similarity module that is called Semsim that measures the similarity between objects (*i.e.* the similarity between two users on the basis of several properties). The recommender methods are based on collaborative or content filtering algorithms. The user data can be represented in java object models, semantic models, or on a Jena RDF Store.

2.3.5.3 MyMediLite

MyMediaLite[3] is an open-source lightweight, multi-purpose library of recommender system algorithms developed on C# for the .NET platform. The data can be loaded from

[1] http://mahout.apache.org
[2] http://gusto.sourceforge.net
[3] http://www.ismll.uni-hildesheim.de/mymedialite

a database but this library also supports multiple file formats (i.e. MovieLens 1M/10M format, ratings, user and item lists, item recommendation files) to load the data in order to make recommendations. In order to build a recommendation system MymediLite can provide collaborative and content recommendation and prediction methods.

2.3.5.4 Gremlin

Gremlin[4] is a graph transversal language that can be used for graph query, analysis, and manipulation. This language is Java based and provides native support for Java, Groovy, and Scala. The data can be loaded from several graph repositories as TinkerGraph, Neo4j, OrientDB, DEX, Rexster, and Sail RDF Stores. Gremlin can rank the nodes near to a given node. This function can be used to develop collaborative and content filtering in order to make recommendations.

2.3.5.5 LinkedData Sail

LinkedData[5] Sail is a particular implementation of the sail interfaces that threats the Web of Data as a single RDF store. This interface allows gremlin to load data from the Web of Data in order to create a graph and make recommendations.

2.3.6 Ready to use Recommender Systems

Ready to use Recommender Systems are autonomous recommender systems with an API to communicate with other applications. The next sub-sections present some Ready to use Recommender Systems.

2.3.6.1 C-IKNOW

C-IKNOW[6] is a semantic recommender system that integrates social network analysis and automated reasoning. Web crawlers, text miners and tagging tools, capture the web data used in recommendation. The recommender process is based on: (i) Geodesic distance, (ii) Positive matches and (iii) Profile similarity. This recommender process returns the same scores for all users based on the search keyword and recommended items followed by a selection stage that incorporates information about the relationship between the user and a potential recommendation to arrive at a final score.

[4]https://github.com/tinkerpop/gremlin/wiki
[5]https://github.com/tinkerpop/gremlin/wiki/LinkedData-Sail
[6]http://ciknow.northwestern.edu

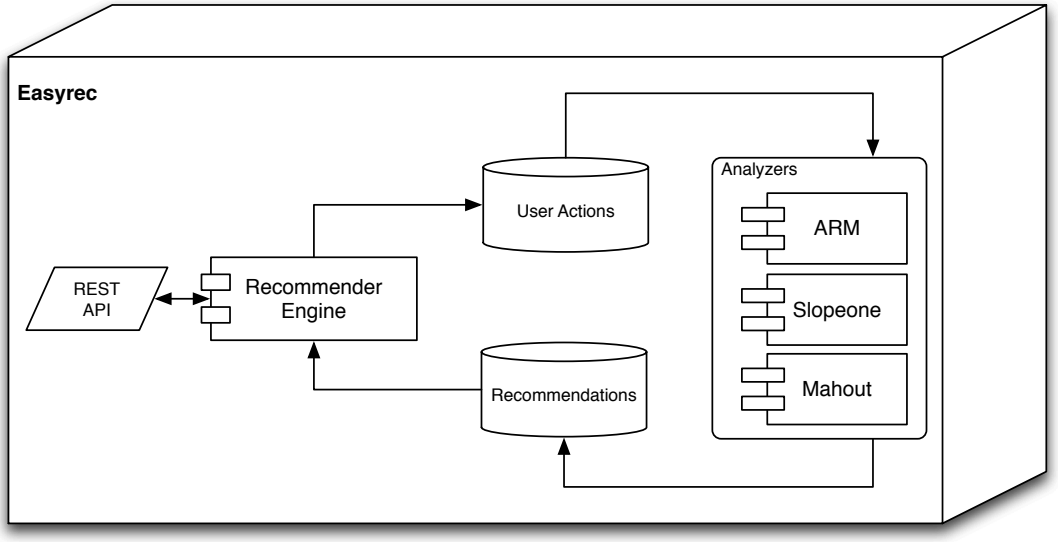

FIGURE 2.8: Easyrec architecture

2.3.6.2 Easyrec

Easyrec[7] is an open source recommender engine that provides recommendations based on user actions. The default user actions are buy, rate and view but more actions can be added. The user and actions information is achieved by a Representational State Transfer (REST) [Ray and Kulchenko, 2002] WebService API and stored in a database. In order to identify patterns to generate recommendations, recommender analysers will periodically analyse this information. Generated recommendations can be requested and accessed through calls to the Rest WebService API and presented to a user.

The Easyrec Recommender System Architecture is presented in Fig. 2.8. It has the following components: (i) Recommender Engine; (ii) a User Actions database; (iii) a set of analysers; (iv) a Recommendations database and (v) a Rest API to input and output data.

Easyrec works based on tenants. Each tenant is a unique identifier of a working project. For each tenant we can have separated items, users, association rules, plug-ins and user actions. The recommendation is based on a collaborative-based filtering that uses association types to define the relation between two items.

The Easyrec analysers can use one of three plug-ins to calculate relations between the users: (i) **Slopeone** - Generates relations based on SlopeOne [Lemire and Maclachlan, 2005] method that analyses item ratings and tries to predict how unrated items would

[7]http://easyrec.org

be rated by the community; (ii) **ARM** - generate rules of the type viewed/bought/-good rated together [Agrawal et al., 1993]; (iii) **Mahout** - Use Apache Mahout to do Collaborative Filtering based on the Apache taste framework.

The recommendations are ranked based in the highest confidence value of the relations between resources but in the case of multiple relations, the recommendation confidence is given by the Equation 2.8.

$$RecConfidence = \frac{\sum MAX(RelationsConfidence)}{nRelations} \tag{2.8}$$

2.3.6.3 OpenRecommender

OpenRecommender[8] is an open source recommender engine that is capable of intelligently retrieving, sorting, ranking, filtering, aggregating and displaying data choices to users. This recommender engine can integrate data from multiple sources as Comma-separated values (CSV), Extensible Markup Language (XML), JavaScript Object Notation (JSON) or RDF formats. The recommender system is based on the Apache Mahout machine-learning library.

2.3.7 Recommender Systems and Libraries Comparison

In order to compare the recommender systems and libraries transversally some functional properties will be considered:

Type To distinguish the ready to use recommender systems from the libraries that can be used to build recommender systems, the property *Type* is used. This property can have the values of *ready-to-use* or *Library*, to define if it is a *ready-to-use* recommender system or a *Library* respectively.

Semantic In order to verify if the recommender system uses semantic data in its recommendations the *Semantic* property is used. Depending on their support of semantic data, this property receives the values *yes* or *no*;

Configuration To evaluate the process of obtaining the distribution of the recommender system or library to have the recommender system *set-up* and running, the property *Configuration* is used. In the case of a *ready-to-use* recommender system the process is composed by an installation and configuration. In the case of the recommender system libraries, the integration complexity to build a simple recommender system must be considered. The installation and configuration of a

[8] http://openrecommender.org

Table 2.2: Recommender Systems and Libraries Comparison

Name	Type	Semantic	Configuration	Documentation
Easyrec	ready-to-use	no	easy	good
C-IKNOW	ready-to-use	no	medium	good
OpenRecommender	ready-to-use	no	medium	poor
Mahout	Library	no	hard	good
GUSTO	Library	yes	hard	medium
MyMediaLite	Library	no	hard	good
Gremlin	Library	yes	hard	good
LinkedDataSail	Library	yes	hard	medium

recommender library is typically more complex than the installation and configuration of a *ready-to-use* recommender system. The values used to measure this process complexity are *easy*, *medium* or *hard* respectively;

Documentation To evaluate the Documentation available for each recommender system/library, the values *good*, *medium* or *poor* are used.

In Table 2.2 a comparison between recommender systems and libraries is presented using the aforementioned properties.

Chapter 3

Architecture Proposal

Recommending access to resources located on different web domains is different and much harder than recommending access to resources on the same domain. In order to recommend access to resources located in multiple domains, resources, access privileges and users must be related. The input data for the process of recommending access for resources located on different domains is originated from a framework proposed by the authors in [Bettencourt et al., 2012] and shortly described in Chapter 2, section 2.1.

In addition to those components, this proposal suggests the inclusion of a fifth component, the Policy Recommendation Point (PRP). The main goal of this component is:

- To predict new useful access privileges based on users, resources, previous access privileges, previous recommendations and traceability information;
- To propose access privileges to the PAP component. The PAP component will later decide if the recommended privilege will be assigned or not based on the decision of the resource's owner or on previous similar decisions.

The overview of the integration of the PRP in the previous framework is presented in Fig. 3.1.

3.1 Policy Recommendation Point

The main component of the PRP is a recommender system that predicts the new access privileges and forwards the recommendation to the PAP component. Accordingly, the recommender system is composed by (Fig. 3.2):

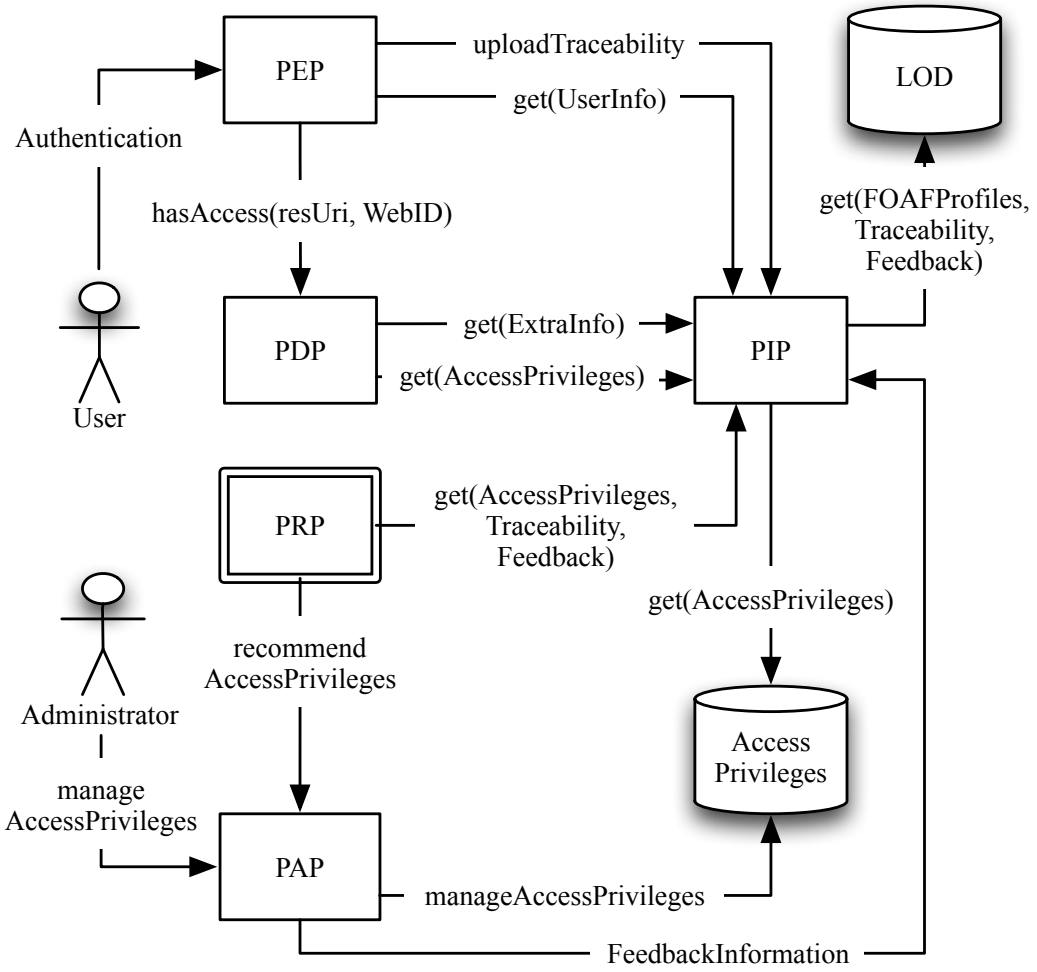

FIGURE 3.1: System architecture including Policy Recommendation Point

- a prediction engine that will predict access privileges based on access privileges, traceability information and previous recommendations feedback,
- a recommender engine that will evaluate the predictions in order to decide if they will be recommended,
- an interface mechanism that request information form the other components, and forwards the recommendations to the PAP.

3.1.1 Data Model

The data to be used by the recommender system is semantically structured and is stored in LOD repositories. The domain data model of PRP is presented in Fig. 3.3.

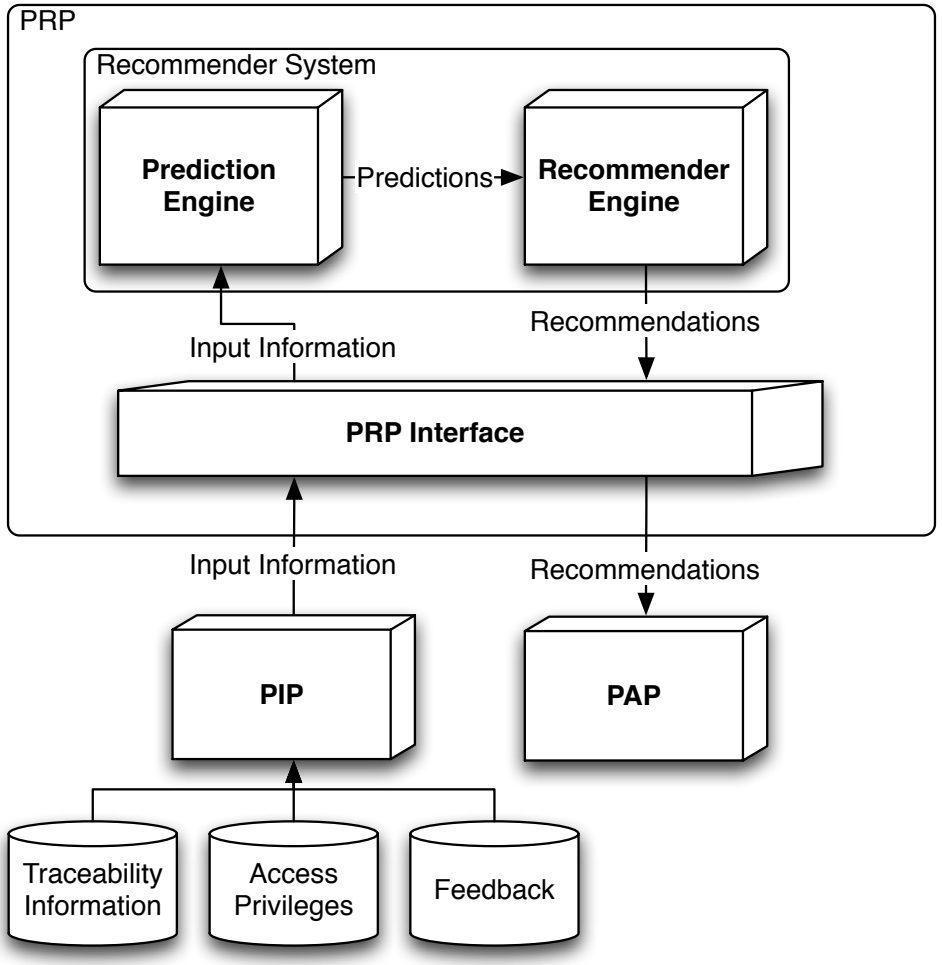

FIGURE 3.2: Policy Recommender Point architecture

Resource is the SuperClass of all concepts and is identified by a URI. The *User* concept has attributes and can be related with other users. To represent access control information, *Privilege* and *Action* concepts must be considered. Privileges have actions over resources that can be performed by users. To represent traceability information the *Activity* concept is used. This concept models the performed actions by users upon resources. The recommendations generated by the PRP are represented by the concept *Recommendation*. Each recommendation has a privilege associated.

Despite the architecture and in particular the recommender system are (must be) agnostic in respect to the concrete adopted data model, several ontologies about the previous concepts are publicly available, thus potentially useful in the instantiation of the architecture:

- The FOAF Vocabulary [Brickley and Miller, 2010] are useful in describing the users' preferences and social network relationships;

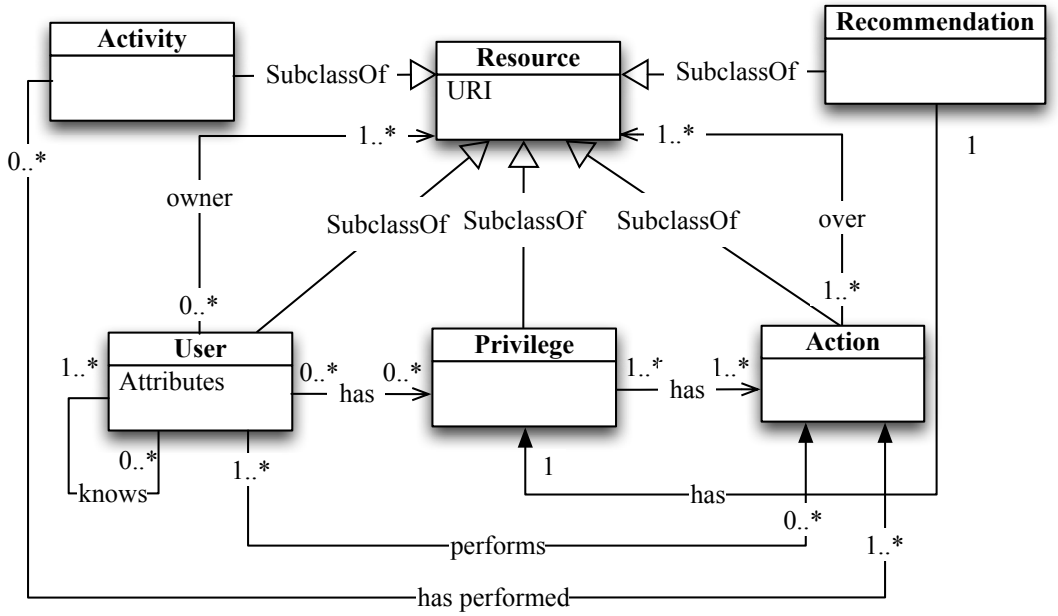

FIGURE 3.3: Policy Recommender Point domain model

- The Provenance Vocabulary [Hartig and Zhao, 2012] is capable to describing traceability information;
- The Dublin Core is capable to describe context information from the users' uploaded resources;
- Several models exists e.g. [Kuhn et al., 2010] and [Finin et al., 2008] to represent privileges.

The output of the recommender system will be semantic-based recommendations of access privileges.

3.1.2 Recommender System

The user actions are low in content information but are very efficient relating (multiple) users with resources. The relations between users and resources through performed actions is showed in Fig. 3.4 where performed actions are represented by the connection arrows.

Traceability information relates users with resources through the actions performed by the users over resources. It is our conviction that this information graph allows inferring user's preferences and trends. In spite of this, a collaborative filtering technique is suggested. With the application of a collaborative filtering technique the prediction engine will be able to predict associations between resources and calculate interests of

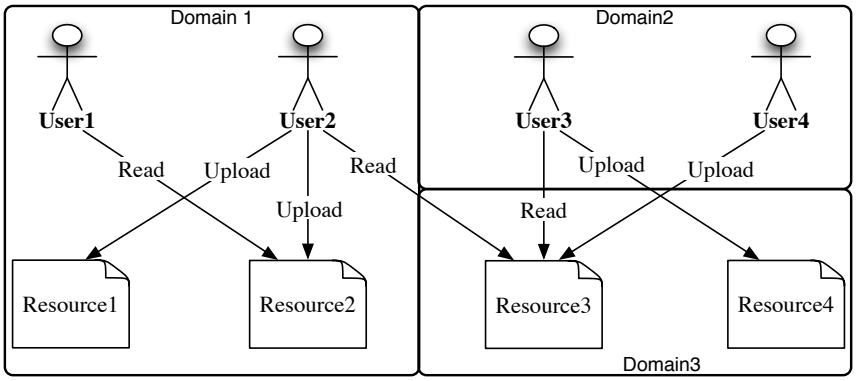

FIGURE 3.4: Relations between users, actions and resources

users in resources based on traceability information. Based on the predicted interests, the recommender engine will evaluate those predictions and associate them to a specific action in order to recommend new privilege. The prediction engine must also consider the feedback information for those predictions and in case of a privilege that has been already accepted or rejected by the resource owner, the recommendation will be discarded. In addition, the recommender engine can also verify if the proposed user has already access privilege to the resource by contacting the PRP. Recommendations representation must be based on a ranked list for each resource owner. The resource owner will only be notified with the top N resources ranked by the prediction score of the proposed user in the resource and some other random resources from the top N+X.

3.1.3 PRP Interface

After the recommendation process, the recommended privileges will be sent to the PAP so the resource owner is notified. Due to the FOAF+SSL authentication approach between the components of the architecture, the PRP Interface is responsible for the communication between PRP and the rest of the system (*i.e.* PAP and PIP). Therefore, this component acts like a wrapper of the recommender system, transforming it into a PRP.

3.2 Recommendation Feedback

After the access privileges recommendation, the PAP will decide if the recommended access privilege will be accepted or rejected. New features must be added to the PAP in order to ensure the creation of the new recommended privileges and to ensure the decision of accepting or rejecting the recommended privilege. This decision making

process can be done (i) explicitly by the resource owner using the PAP user interface or (ii) inferred (implicitly) from the user actions. In Fig. 3.5 it is present the flow diagram respecting this process.

In order to support this process, changes are suggested to the PAP component of the original system framework. In particular, the PAP component will be enhanced with inference/reasoning capabilities. Fig. 3.6 presents the recommendation feedback components in the overall system architecture.

3.2.1 User explicit feedback

Any user, provided that is uniquely identified by a FOAF profile and has a valid FOAF+SSL authentication, is able to receive access recommendations by connecting to a PAP user interface. To ensure the explicit feedback, after receiving the recommendations from the PRP, PAP will provide to the resource owners an interface to list, accept or reject the recommended access privileges. When a resource owner accepts the recommended privilege, the PAP will act in order to: (i) assign the privilege to the proposed user; (ii) register in the PIP that the access privilege was accepted and (iii) notify the proposed user that a new recommended resource is available. When the resource owner rejects the recommended privilege, the PAP will register that the access privilege was rejected in the PIP. In both cases (acceptance and rejection) the PIP will store feedback information in order to be used in a future recommendation process.

3.2.2 Inferred feedback

In the case of the privilege recommendation feedback (*i.e.* acceptance or rejection) is done based on the resource owner explicit decision, PAP will inquire the resource owner if s/he wish to accept or reject the recommended privileges. After getting the owner feedback, the PAP component must act accordingly to the feedback and send that information to a PIP. In order to infer the decision from the user previous actions, a rule-based reasoner is used. Typically, the rules will consider the previous user explicit decision and the characteristics of the recommendation. For example, a recommendation will have the same feedback as a similar recommendation for the same/similar user. This implicit feedback together with the assumption that users will have a coherent behaviour, thus making the same decisions in the future, can be used to prune access recommendations whose similar explanation had been accepted or rejected before. This inferred feedback releases the user of the burdensome of having to accept or reject several similar predictions in the future.

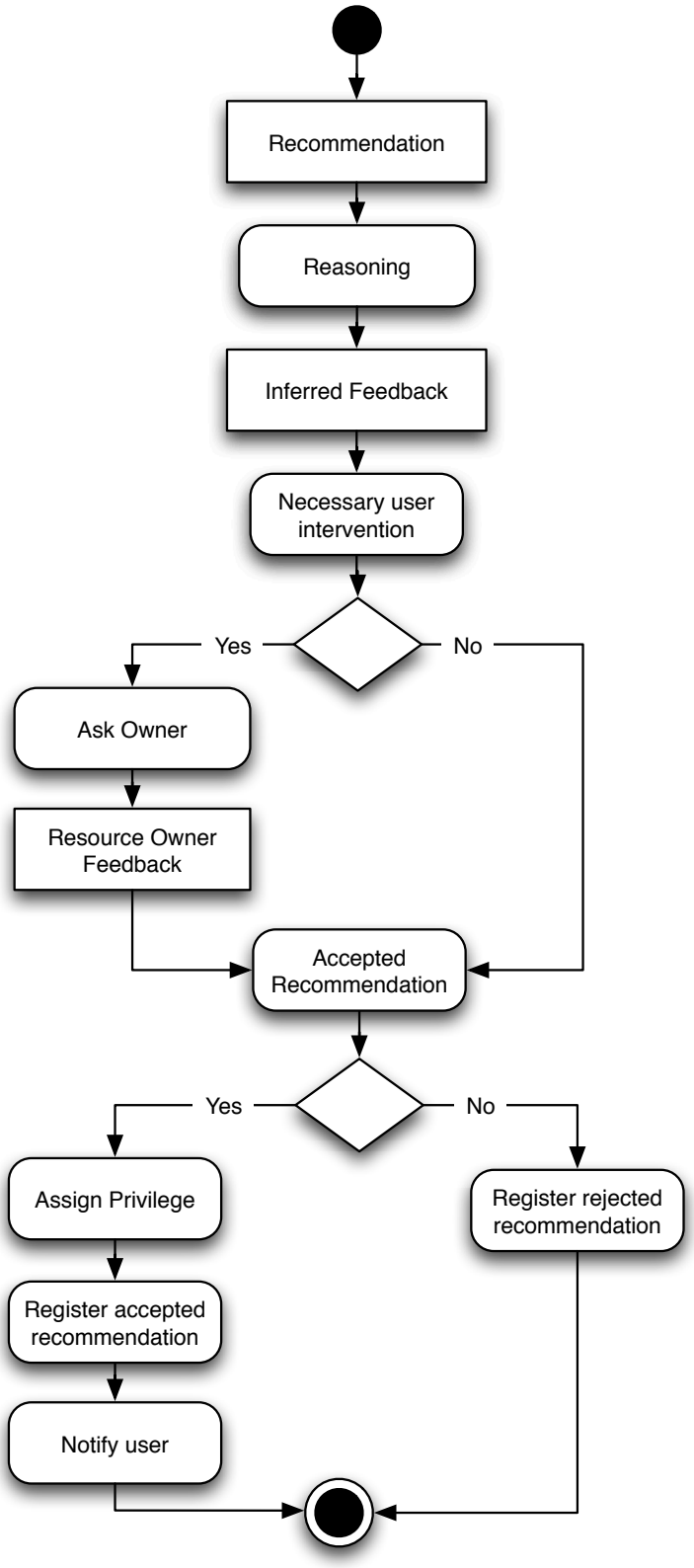

FIGURE 3.5: PAP owner feedback flow diagram

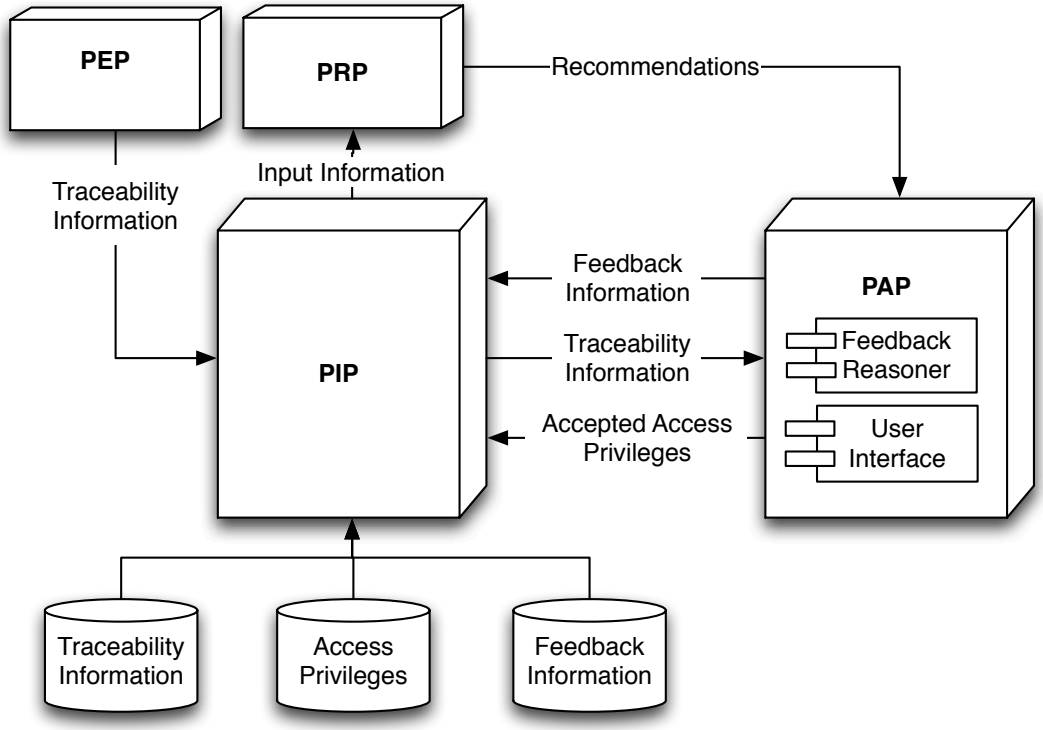

FIGURE 3.6: The recommendation feedback in the perspective of the system architecture

3.2.3 Summary

This chapter proposed the inclusion of (i) a PRP to the previous access control framework, (ii) its data model and (iii) an interface to communicate with the previous components. It was also proposed how to capture recommendation feedback based on user explicit feedback or inferred feedback. As this proposed architecture cannot be formally evaluated, the next chapter describes its instantiation in order to be experimented and evaluated.

Chapter 4

Architecture Instantiation

This chapter describes the instantiation of the proposed architecture enhancements. Three subjects had to be considered:

- The data model and data management process;
- The implementation of the PRP;
- The deployment of the PRP in scope of the rest of the system.

As described in the architecture proposal, the access control platform proposed by [Bettencourt and Silva, 2010] does not meet all the requirements to instantiate the PRP without changing some of its components. Some changes need to be made to the PIP and PAP components in order to ensure its coexistence with the PRP.

4.1 Data Model and Management

In the proposal of the automatic traceability acquisition framework [Bettencourt et al., 2012], no specific ontology was defined to represent traceability information. In order for generic recommender systems to make use of existing access control privileges to recommend other access privileges, the information must be represented in an simple format without relying on complex ontologies but at the same time extensible enough to be used by generic access control systems. As a result, a simple ontology was developed to represent traceability information to be used in the PRP as input data. These access privileges will allow the PRP to not recommend already assigned access privileges because the feedback information that is inserted in the recommender system is based on the access privileges that are accepted or rejected by the resource owner.

4.1.1 Traceability Information

The aim of this work is to recommend access privileges to users based on the traceability information of user actions so the traceability information must be stored and formally represented. To store this traceability information Ontologies and Semantic Web technologies are suggested. In [Hartig and Zhao, 2012] the authors propose an ontology called Provenance Vocabulary Core Ontology to represent provenance of Web data. However this ontology cannot represent the performed actions over resources (*i.e.* upload, read, write, download). To overtake this limitation the ontology has been extended. Because the Provenance Vocabulary Core Ontology is an extension of the ontology PROV-O it uses two namespaces, one for the representation of concepts from the Provenance Vocabulary Core Ontology (prv)[1] and other to represent the concepts from the The PROV Ontology(PROV-O)[2] with the namespace (prov) .

From the Provenance Vocabulary Core Ontology the most relevant concepts to be used to represent the traceability information are:

- *prv:DataAccess*: DataAccess is a subclass of *prov:Activity* and represents the completed execution of an activity by which a data item has been retrieved;
- *prv:DataCreation*: DataCreation is a subclass of *prov:Activity* that represents the execution of an activity by which data items have been created;
- *prv:DataItem*: DataItem is a general concept that represents data items of any kind;
- *prv:HumanAgent*: HumanAgent is a general class that represents agents who are social beings (*i.e.* persons, organizations or companies).
- *prv:accessedResource*: This property refers to the Web resource that has been accessed during the execution of a data access;
- *prv:completedAt*: This property refers to the time an activity has been completed;
- *prv:performedBy*: This property refers to an agent that/who performed an activity;
- *prv:createdBy*: This property refers to the creation of a data item.

[1] http://purl.org/net/provenance/ns
[2] http://www.w3.org/ns/prov

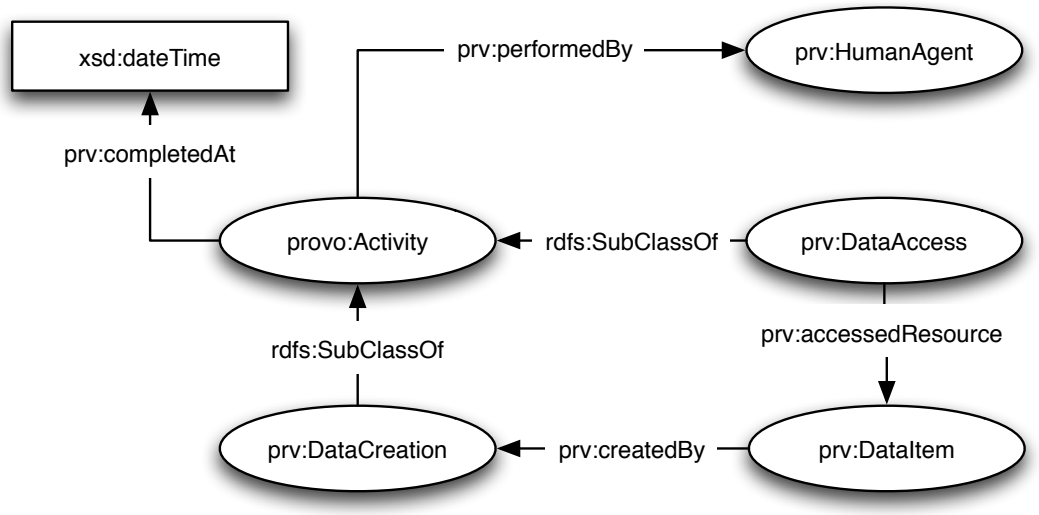

FIGURE 4.1: Using Provenance Vocabulary Core ontology to represent access traceability information

Activities performed by users are represented by the *provo:Activity* class of the PROV Ontology (PROV-O). This class is extended in the Provenance Vocabulary Core Ontology through the *prv:completedAt* property in order to represent the activity completion date time. The *provo:Activity* class is also extended by the *prv:performedBy* property that will relate the activity and who performed the activity. Who performed the activity will be represented by an agent, in this case an human agent represented by the *prv:HumanAgent* class. The Provenance Vocabulary Core Ontology defines two types of activities, access activities and creation activities.

Access activities are actions performed by a user over an item that does not change the item (i.e. Read actions). The concept *prv:DataAccess* will be used to represent access activities and will be related with the accessed item through the *prv:accessedResource* property. Creation activities are those where data items have been created (*i.e.* Upload/Editions actions). The *prv:DataCreation* concept is used to represent creation activities and will be related with the created item by the *prv:createdBy* property. The *prv:DataItem* class is used to represent resources.

However, the Provenance Vocabulary Core Ontology cannot define the type of actions performed over a resource. To overcome this limitation we extended the Provenance Vocabulary Core Ontology. The new ontology has the prefix "prva" and will add a new class to represent actions (*i.e.* read, write, upload) called *"prva:Action"* and a new property called *"prva:performedAction"* to relate the action with the activity as presented in Fig. 4.2. For the Provenance Vocabulary Core Ontology to be compatible with the users FOAF profiles, the *foaf:Person* class is extended as a subclass of class

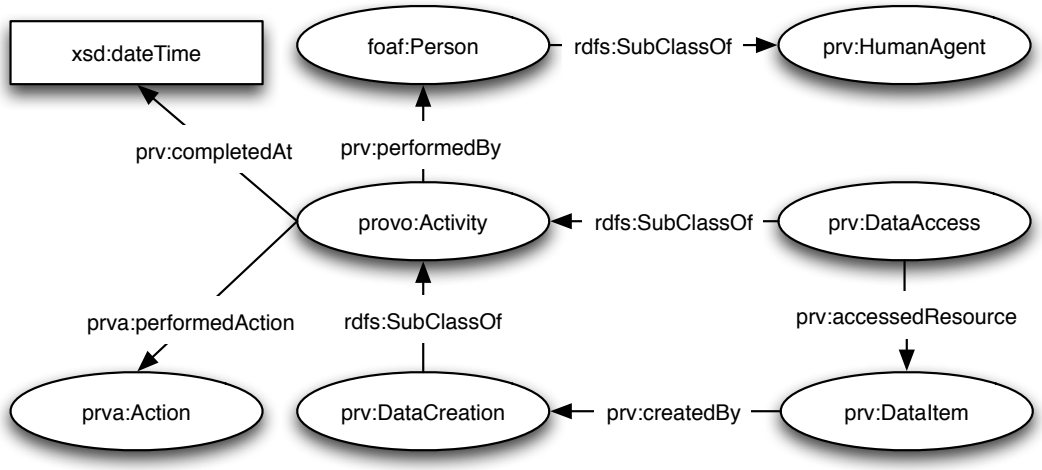

FIGURE 4.2: Extended Provenance Vocabulary Core ontology to data access

prv:HumanAgent. As consequence, the use of FOAF+SSL authentication is mandatory (natively supported by the original access control platform).

4.1.2 Access Privileges

In order to be used by generic recommender systems, input access privileges to PRP will be represented by a simple ontology. This ontology goal is to represent generic access privileges and relate them to actions, resources and persons in order to be easily extended by more complex access control ontologies as Role Based Access Control in OWL (ROWLBAC) [Finin et al., 2008] or the ontology presented by the authors in [Kuhn et al., 2010]. This ontology is presented in Fig. 4.3 and will be represented by the prefix *"priv"*. Access privileges will be represented by the class *priv:Privilege*. To represent users, this ontology uses the class *foaf:Person* of the FOAF ontology as an extension. Users will be related to privileges through the property *priv:hasPrivilege*. Resources are defined by the class *priv:Resource* and are related with privileges through the property *priv:overResource*. The actions are represented through the property *priv:Action* and related with privileges by the property *priv:allowedAction*.

4.1.3 Feedback Information

The recommendation items that will be recommended by the policy recommender point are (changes to) access privileges. Access privileges are represented by the ontology presented in Fig. 4.3 but this ontology must be extended in order to represent feedback information. In such cases only the owner privilege acceptance or rejection feedback will

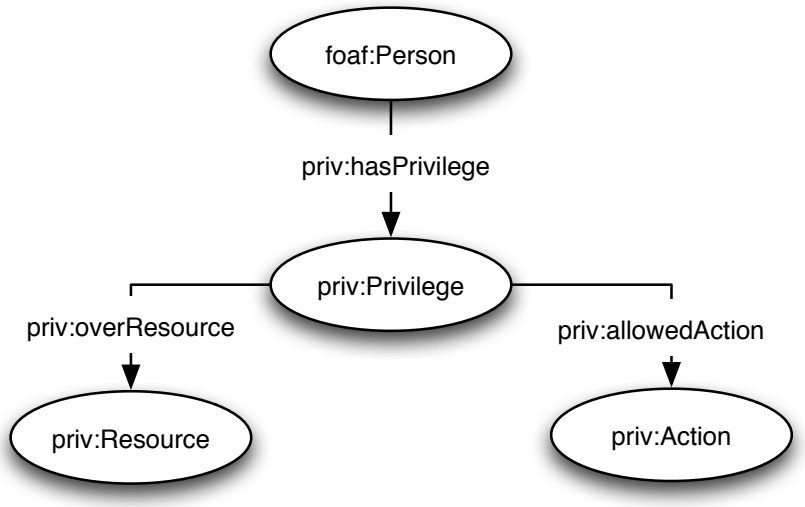

FIGURE 4.3: Extended Provenance Vocabulary Core ontology

be considered. This information will be accessed by the PIP in order to provide it to the PRP that will use it in future recommendations.

4.2 Policy Recommender Point Instantiation

As described in the architecture proposal, the instantiation of the PRP is done by the adoption of a third-party collaborative-based Recommender System that uses user actions to predict user interests and recommend changes to access privileges. The Easyrec Recommender System was selected because it was designed using collaborative filtering to predict resources based on the actions over resources. In order to recommend access privileges based on traceability information, some additional features are necessary. As traceability information is based on actions performed over resources, it is possible to insert traceability information into the Easyrec system through data mapping and transformation.

Nevertheless, as Easyrec's api does not allow new action types or new rule types, but instead they must be pre defined in the user interface provided by the system management HTML configuration application. Also the Easyrec Database is a simple SQL database and does not allow explicit representation of rich semantics of the items or users. In such situations external components are required to deal with the data and its transformation into Easyrec readable, formatted and consistent data. As shown in Fig. 4.4, it is proposed a Semantic importer component to import the traceability information to the Easyrec to ensure data consistency.

After obtaining traceability information, Easyrec plugins infer new relations through predefined rules. The recommendations can be obtained for each user and a trust value on the recommendation is also retrieved. Easyrec can predict the interest of a user in a resource by collaborative filtering plugins but cannot predict if the resource owner wishes to give such access to that user. To overtake that barrier, additional components to predict the interest of the resource owner in assigning the access privilege can be used. Yet, this subject is out of the scope of this thesis. Easyrec recommendations are serialized in XML and can be accessed through the REST API but only for one user at a time. Since those recommendations are made for the creation of access privileges, the recommendation will be done to the resource owner and not to the user that has interest in the resource. To get all recommendations and group them in a ranked list it is proposed an additional component called "Privileges Recommender".

4.2.1 Semantic Importer

The Easyrec recommender system only accepts actions performed by users over resources as input. The PRP must recommend access privileges based on traceability information, feedback and previous access privileges. In order to transform this input information into actions and send them to Easyrec, three sub-components are developed and presented in Fig. 4.5. However, Easyrec does not recommend resources to users that have already performed an action over those resources. One way for Easyrec to do not recommend resources that the user already has access is to transform access privileges into actions. As feedback information is related to accepted or rejected privileges, it is possible to transform them into actions and import them alongside, does serving as prediction input and therefore constraining the prediction process. The first component called "Privileges and feedback transformer" has the task to transform previous access privileges and feedback information in traceability information. This traceability information will be transformed into Easyrec readable actions by the "Traceability actions adapter" and sent to the Easyrec REST API by the "Easyrec REST communication Interface" sub-component.

4.2.2 Privileges Recommender

Easyrec provides recommendations serialized in XML for each user through its REST API. These recommendations are based on the interest predicted between the users and resources, but access privileges must be recommended to resource owners and not to the users that might gain access. To ensure that users does not have already access

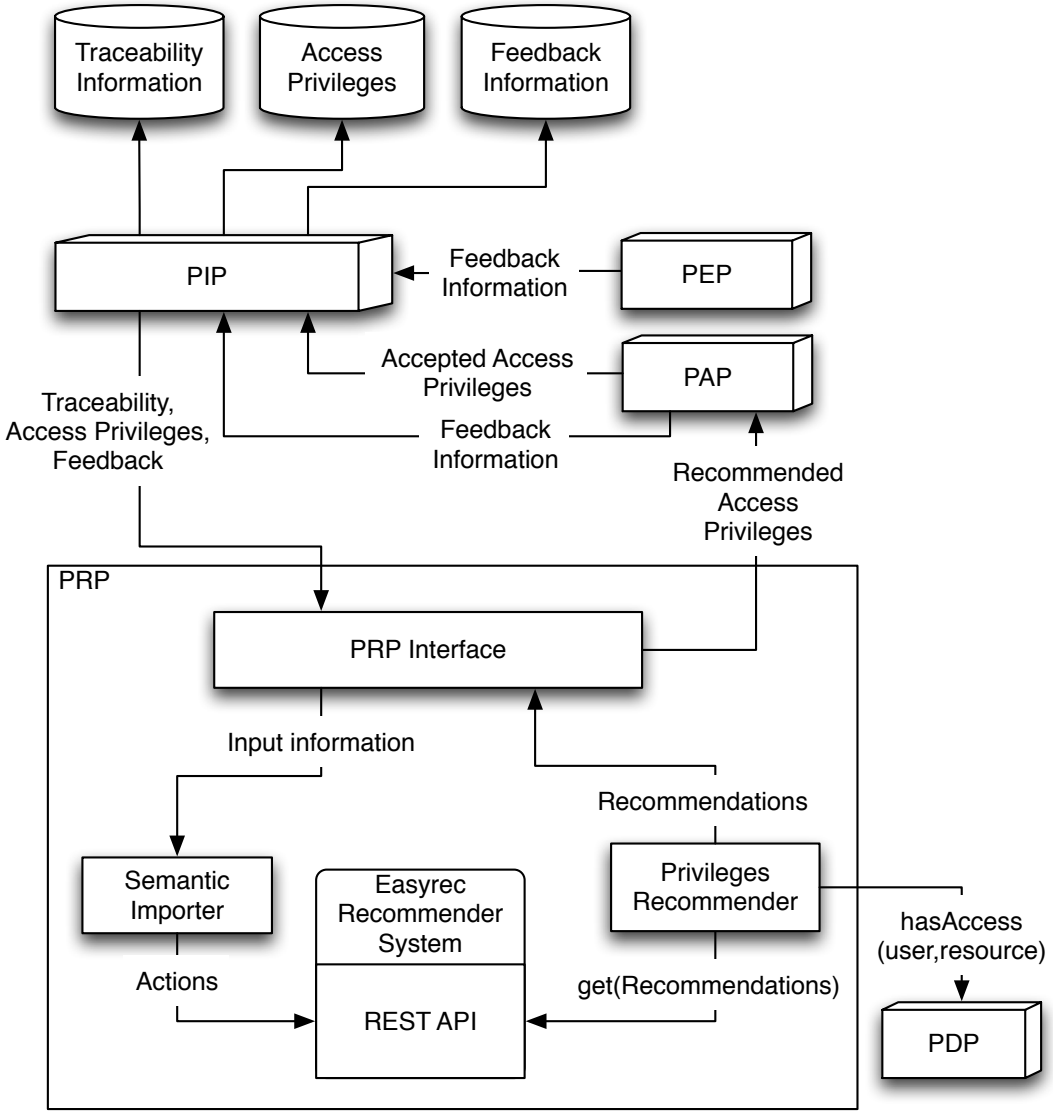

FIGURE 4.4: Using Easyrec to recommend access privileges

privilege to the resource, the PDP must be contacted. To overcome these limitations, the Privileges Recommender has four sub-components as presented in Fig. 4.6:

The Recommendations Importer sub-component will perform a recommender request for each user and get those recommendations.

The Access Verifier component will contact the PDP through an agent capable of FOAF+SSL authentication and verify for each recommendation if the recommended user has already access to the recommended resource. If the user already has access to the resource, the recommendation will be discarded.

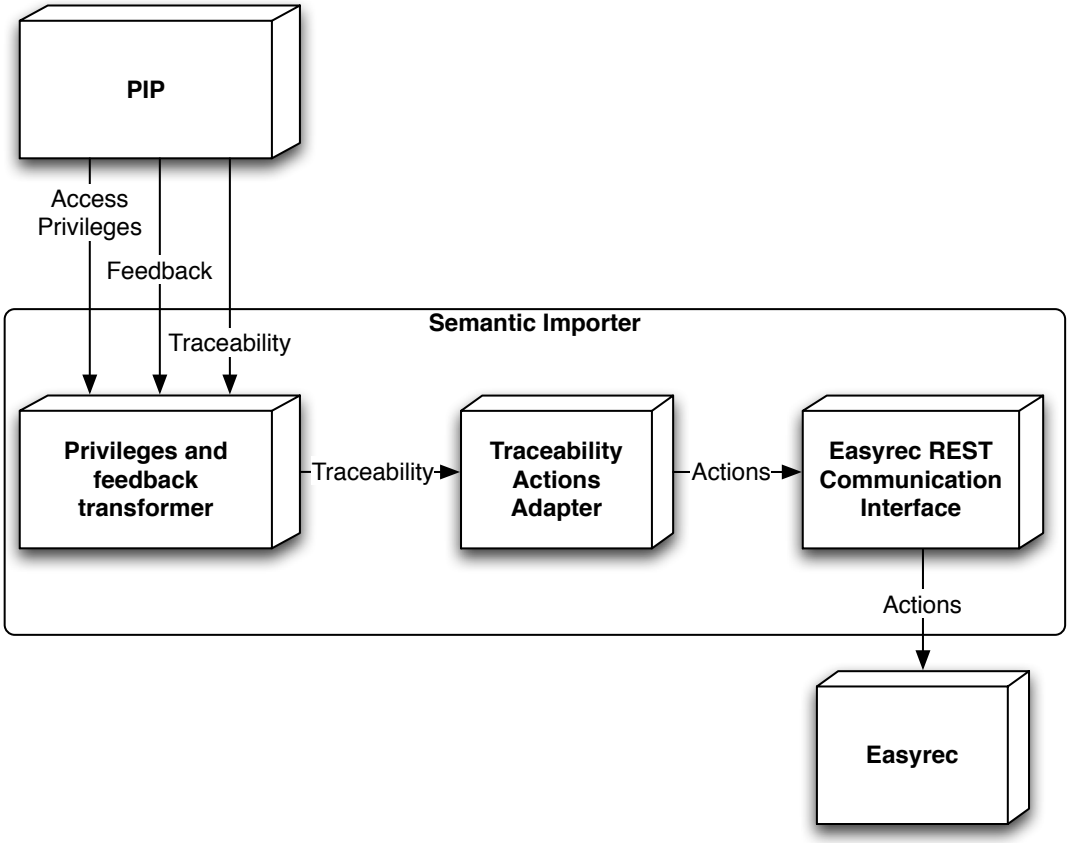

FIGURE 4.5: Semantic importer architecture

The Owner Retriever sub-component will contact PIP for each resource in order to retrieve the resource's owner. The resource's owner will be added to the recommendation list and sent to the Privilege Creator sub-component.

The Privilege Creator sub-component will create access privileges with the recommendation information and send them to the PRP Interface sub-component that will send the recommended privileges to PAP.

4.3 Policy Recommender Point Deployment

As requested the PRP should be deployed on web servers that have the capability to communicate with: (i) the PAP in order to recommend access privileges and (ii) the PIP to retrieve users, resources and other meaningful information. These components do not necessarily have to be in the same Web Server but must be accessible by HTTPS with FOAF+SSL authentication. For simplicity reasons, we assume that the PRP and PAP components are deployed on the same Web Server as in Fig. 4.7, but they can coexist on different Web Servers.

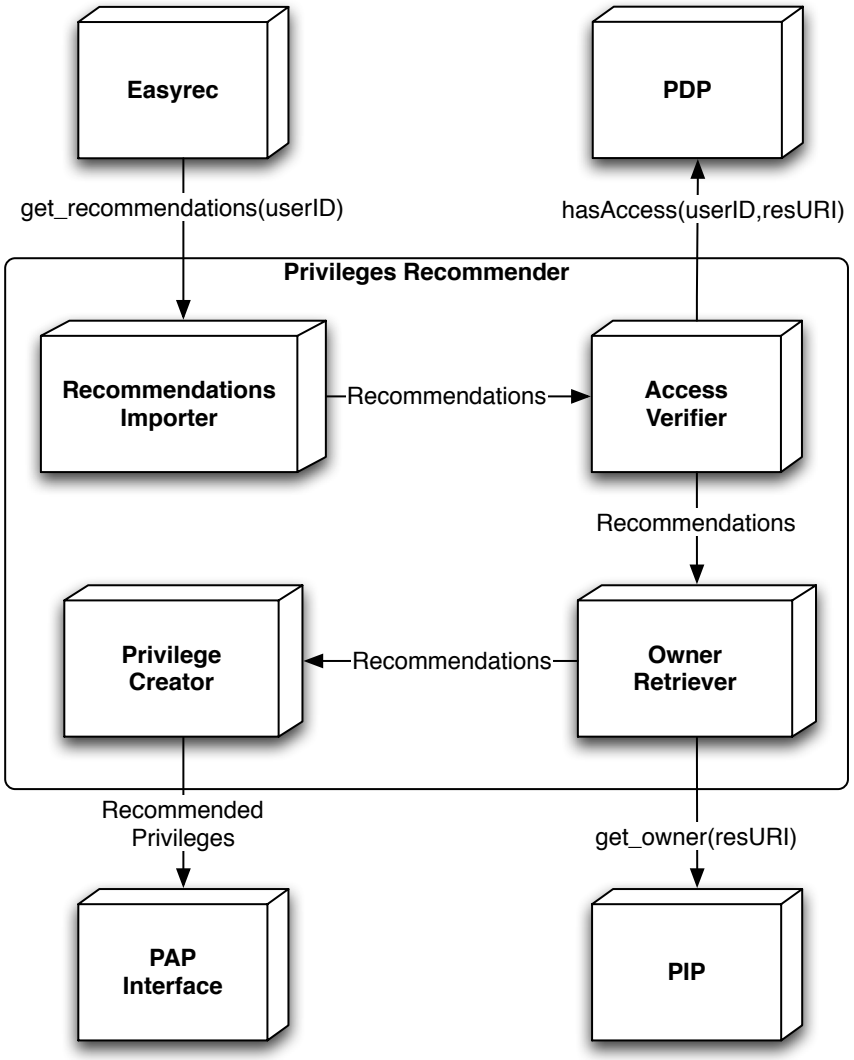

FIGURE 4.6: Privileges Recommender architecture

As show in Fig. 4.7, distinct applications on distinct Web servers can generate traceability information. This information will be used to recommend the access privileges across domains.

As in the communication between the PEP and the PIP, the agents with FOAF+SSL authentication capabilities must ensure the communication between the PRP and the other components.

4.4 Summary

This chapter described the implementation of the PRP using the third-party Easyrec recommender system and its deployment in the scope of the rest of the system. It also

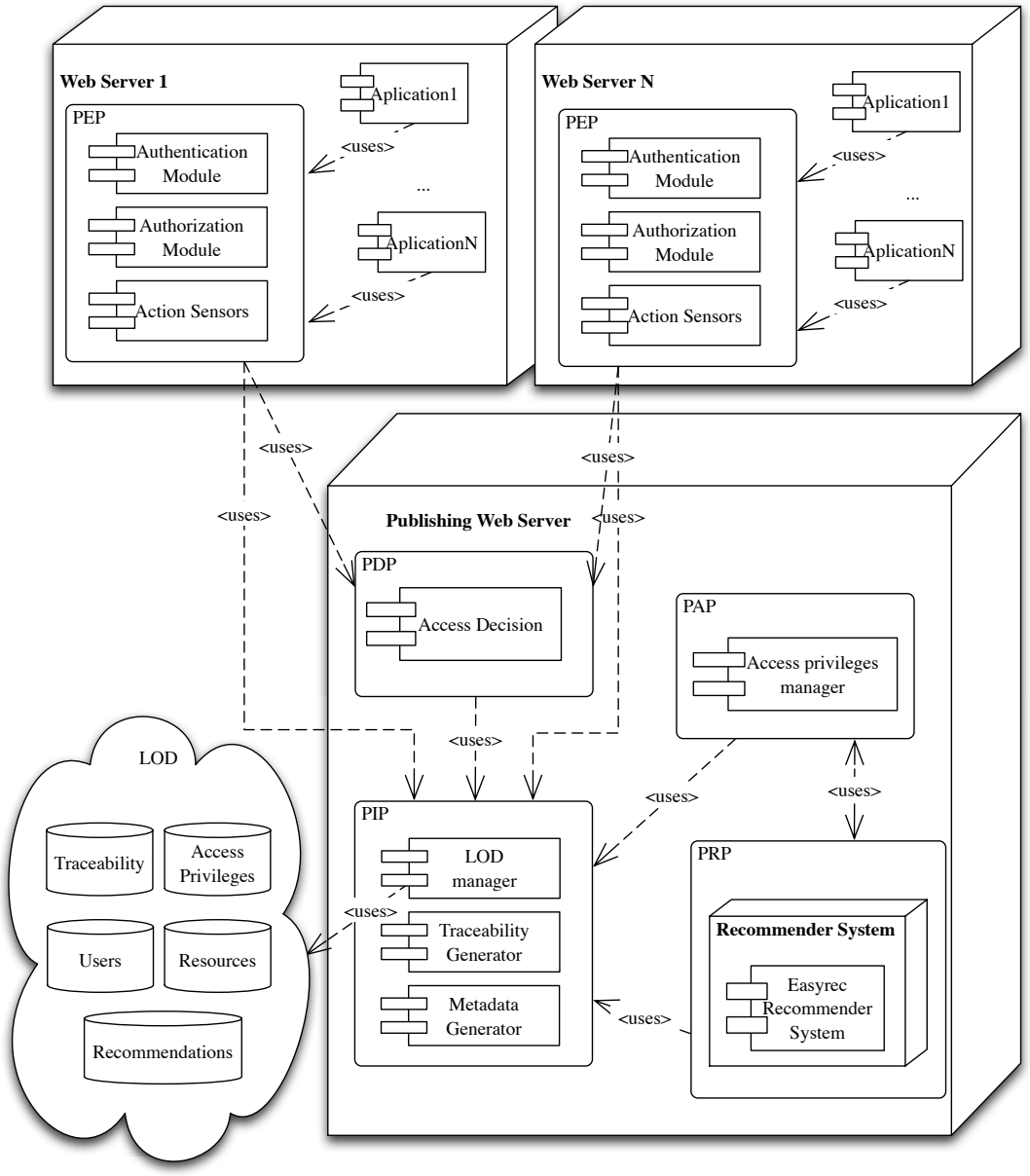

FIGURE 4.7: Policy Recommender Point deployment diagram

described the concrete data model specification using extended/modularized ontologies of access privileges, traceability and feedback information. This architecture instantiation allows the architecture validation through experiments and evaluations with real data described in the next chapter (Experiments).

Chapter 5

Experiments

In order to evaluate the instantiated system, two efforts were necessary:

- Set-up of the recommender system in the scope of the PRP and considering the application domain in hands (*i.e.* access privilege recommendation to web resources considering traceability information);
- Perform experiments for evaluation, with real data from real users. For that, two scenarios were devised:
  - without traceability information, which will represent the reference results;
  - with traceability information that feeds the recommender system. The results will compare with the reference results.

The next section presents the isolated set-up and configuration of easyrec. The second section describes the experiments and evaluation of the overall system.

5.1 Easyrec Set-up

The traceability information relates users and resources only. In order to relate resources with other resources the Easyrec Recommender System infers relations from the actions performed by users through predefined rules. These rules are associated with the actions type in a relation of N to 1 respectively.

Traceability information can have distinct action types so the Easyrec can be predefined with several distinct rules to infer distinct resource relations. These rules are captured in plug-ins so one can use different rules. The plug-in used in this instantiation and

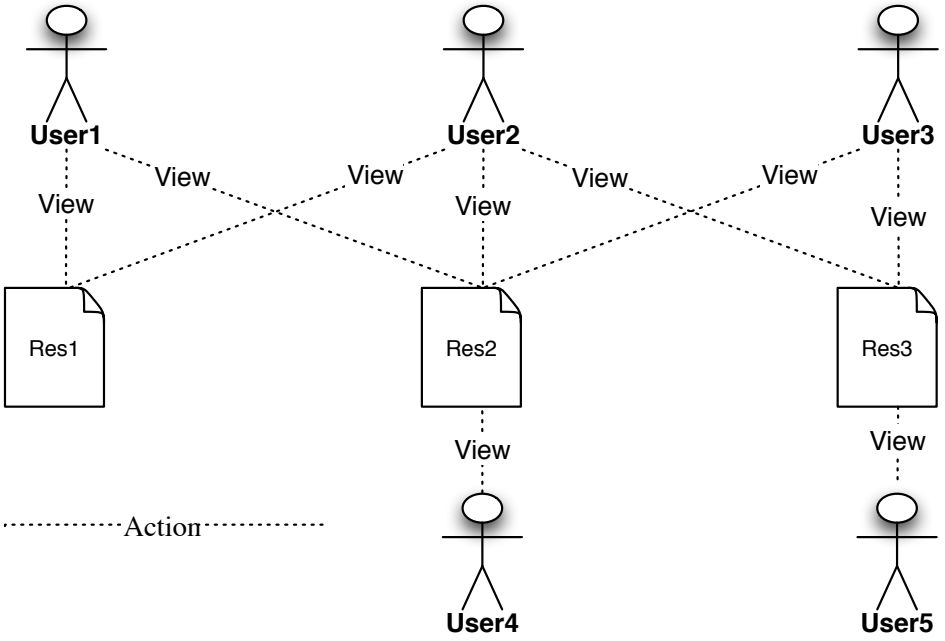

FIGURE 5.1: Easyrec case scenario

experiments to compute relations between resources is the ARM plug-in [Agrawal et al., 1993] that uses confidence to rank items. Confidence describes the likelihood that an item Y follows in the presence of item X. To calculate confidence, Easyrec uses the equation described in Equation 2.5 where support is defined by how often a set of items <x,y> appear together in different user history. While the following section demonstrates the adoption of the Easyrec and the ARM plug-in in this scope of application domain, the second section describes the extensions introduced in Easyrec to accommodate the specificities of the domain application.

5.1.1 Adopting Easyrec and ARM

Consider the case scenario depicted in Fig. 5.1 five users and three resources are considered: the User1 and User3 view two Resources (Res1 and Res2), the User2 views 3 Resources (Res1, Res2, Res3). User4 views the Resource Res2 and the User 5 views Resource Res3.

5.1.1.1 Predict relations

As in the previous case study, an ARM plug-in is used to infer new relations with its default configuration. Support values obtained with ARM plug-in are presented in Table 5.1.

TABLE 5.1: Support values

Resources/pairs	Support (0 - N, Users)
Res1	2
Res2	4
Res3	3
<Res1,Res2>	2
<Res1,Res3>	1
<Res2,Res3>	2

TABLE 5.2: Confidence values

Pair	Confidence(0 - 1)
Res1->Res2	1
Res2->Res1	0,5
Res2->Res3	0,5
Res3->Res2	0,67

By default, the minimum support is 2 so resources/resource pairs with less than that are ignored (*i.e.* resource pair <Res1, Res3> has a support of 1, so it will be ignored). To the other pairs, the Easyrec Recommender System will calculate the confidence based on the Equation 2.5 and using the values obtained in the Table 5.1. Results are presented in Table 5.2.

After calculating the confidence, the Easyrec Recommender System will select the top N (default is 50) resource pairs with best confidence. The confidence for the resources pairs <Res1, Res3> and <Res3, Res1> is not calculated has they do not present enough support and only four relations will be created based on the defined rule (Viewed-Together) with the confidence value associated to them as presented in Fig. 5.2.

5.1.1.2 Recommendations

The Recommender component of the Easyrec Recommender System, will use the relations inferred by the analysers on the previous section as presented in 5.2 to predict the interests of User4 and User5 as presented in Fig. 5.3. The recommender will use the confidence values of the relations to predict and recommend the resources to users. In this case the relation values used to predict the interests of User4 are the confidence values of the pairs with the resource Res2 in the left argument. These pairs are: <Res2, Res1> and <Res2, Res3>.

When we retrieve recommendations to User4 we are presented with recommendations of Resource 1 and Resource 3 with a confidence value of 50.

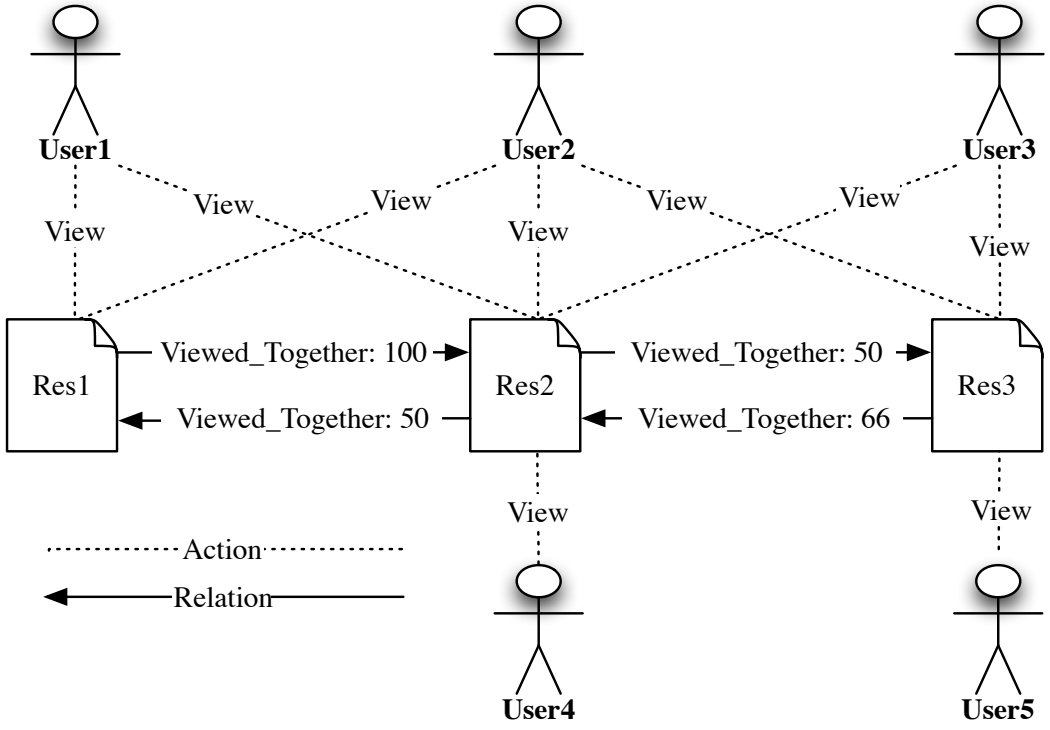

FIGURE 5.2: Infered relations

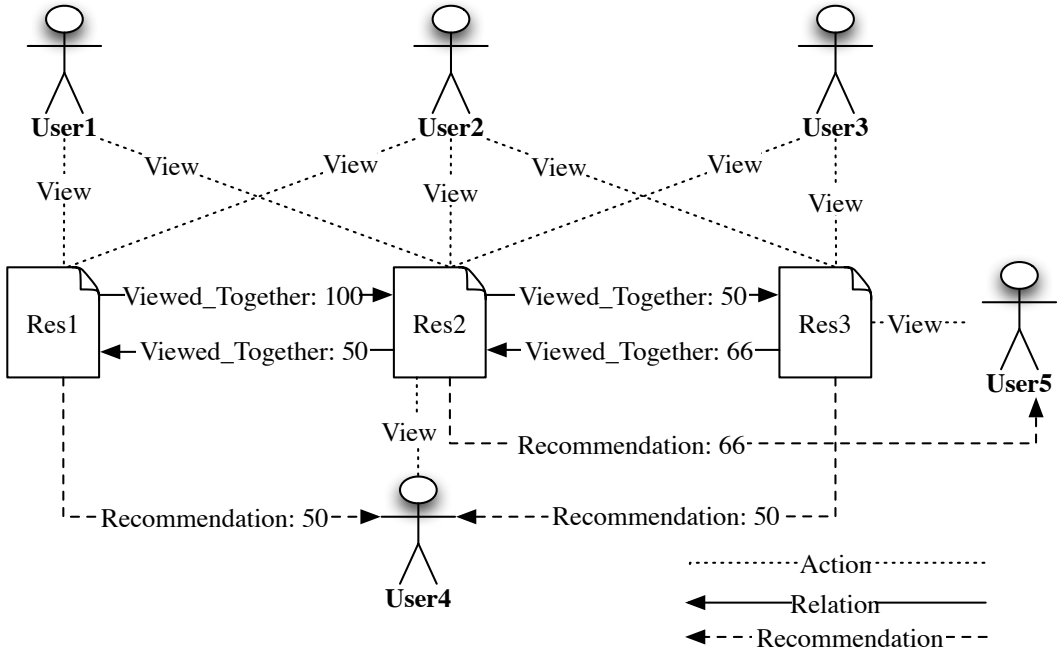

FIGURE 5.3: Recommendations for case scenario

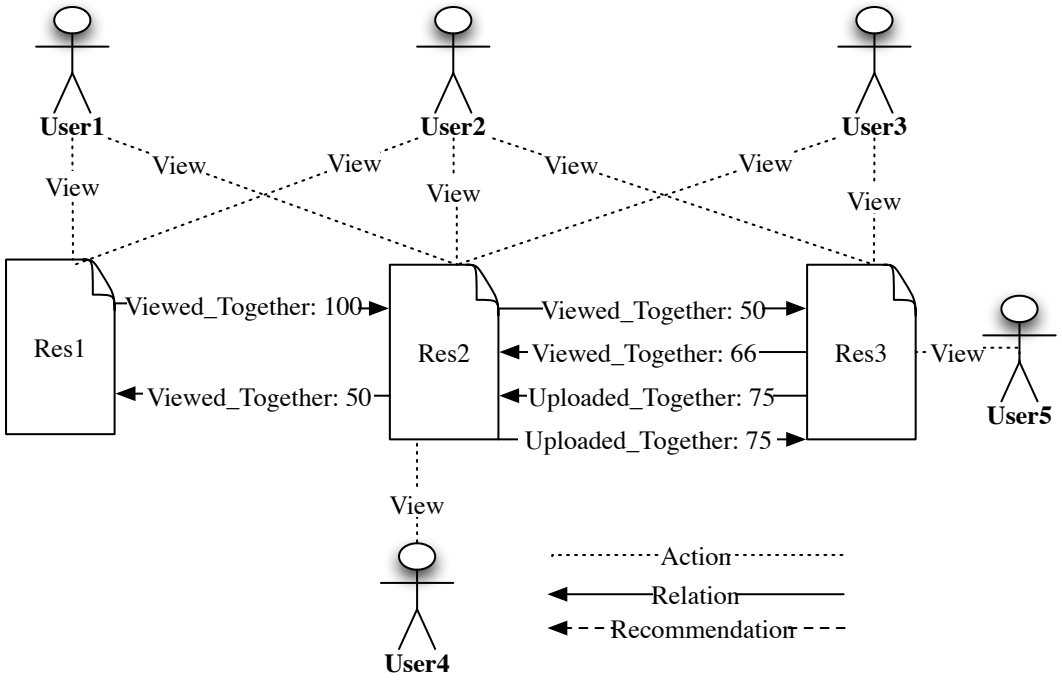

FIGURE 5.4: Case scenario with the relation uploaded_together

To predict the interests of User5, the recommender component will use pairs with the resource Res3 in the left argument. In this case only the pair <Res3, Res2>is found. So only one resource is recommended, that is the Resource 2 with 66,6(6) of confidence.

5.1.2 Extending Easyrec Native Relations

This study will allow evaluating how distinct relations will affect the final recommendation. To the previous case scenario two new relations are added with the type of uploaded_together and confidence value of 75. This relation has the objective of relate resources that have been uploaded by the same person. This relation is not a native Easyrec relation. This relation must be added to the Easyrec recommender system and will represent resources related by resource upload actions. Two relations will be asserted between User 2 and User 3 as presented in Fig. 5.4.

As we have two distinct relations between the pairs <Res2, Res3>and <Res3, Res2>the recommendation component will calculate the final recommender confidence based on the Equation 2.8.

The recommended resources for each user with the confidence value is presented in Table 5.3.

TABLE 5.3: Case study scenario recommendations

		Users	
		User4	User5
Recommendation	Resource1	50	-
	Resource2	-	70,5
	Resource3	62,5	-

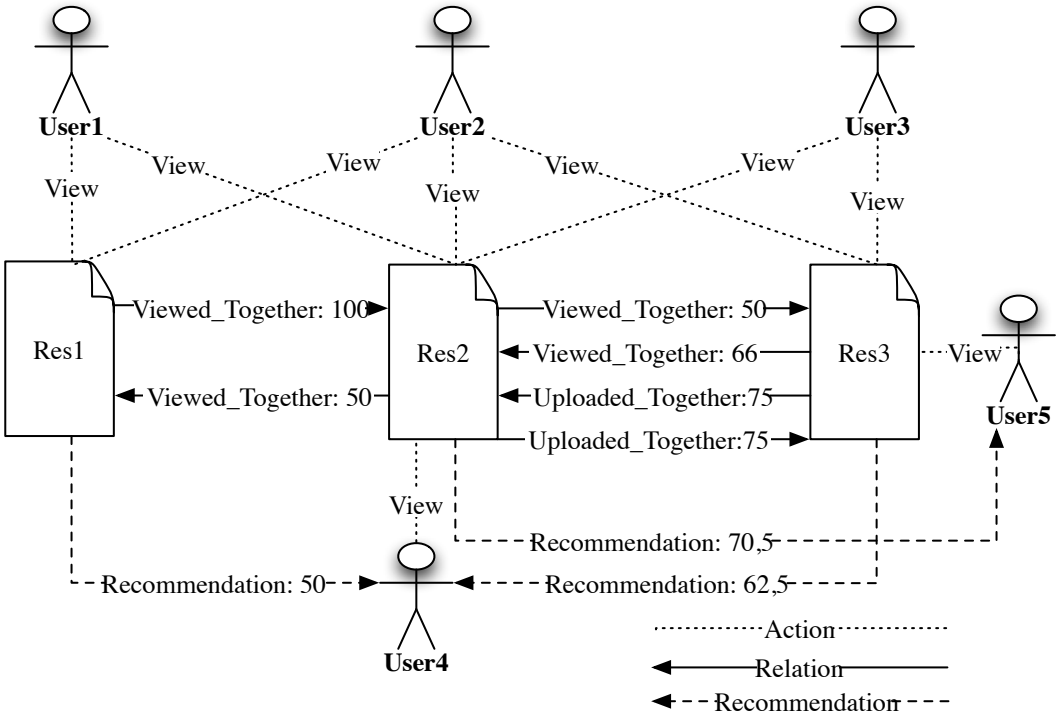

FIGURE 5.5: Recommendations with the relation uploaded_together

5.2 Traceability improvements experiment

To prove that traceability information improves the recommendations of access privileges to users, two experiments have been carried out: (i) one without traceability information and (ii) another with traceability information. Because testing the system in real-time would be overwhelming in terms of time span, a more pragmatic approach was adopted, by considering the browsing history of users, such that information is interpreted as upload and view actions of items (*i.e.* the browsed URI are interpreted as resources). The system will recommend changes to the access privileges of these URI. To evaluate if those predicted resources are useful afterwards, it was done a user survey where users had to answer for each prediction if the resource is useful or not. To evaluate if the recommendations are useful, each user should accept the recommendation or reject according if they want the interested user to be given access or not to the resource.

TABLE 5.4: Set-up data

User	User Type	Number of actions
User1	Common-User	7789
User2	Common-User	11318
User3	Computer Science student	9927
User4	Common-User	16330
User5	Computer Science student	31461
User6	Researcher	19639
User7	Researcher	9240

5.2.1 Set-up

To perform these experiments, seven users provided their browser history. The users can be categorised into researchers, computer science students and common users. The user type and the number of visited resources is presented in Table 5.4.

A key set-up process is the interpretation of the users' browsing history into traceability information, *i.e.* upload, update, view actions, as well as ownership and authorship relationships between users and resources. The adopted interpretation is as follows:

- The first access (read) action upon a resource is interpreted as the traceability upload action. Accordingly, the user is considered the owner of that resource. The access privileges recommendation will only be forwarded to the owner of the resource;
- The access (read) actions upon the same resource occurring within the week after the upload action are interpreted as write/update actions. Hence, the user performing these actions is considered an author of the resource;
- The remaining access (read) actions upon the same resource are interpreted as view actions.

5.2.2 Access recommendation without traceability

The access recommendation without using traceability information is simulated by considering the resource ownership and authorship data only. This set of relations allows simulating collaboration between users. These actions will be inserted directly in the Easyrec recommender system as depicted in Fig. 5.6.

Based on the upload and update actions, Easyrec will compute resource relations of the type "uploaded_together". These relations will be used to predict the user interest about items. The Easyrec overall statistics without traceability are presented in Table 5.5.

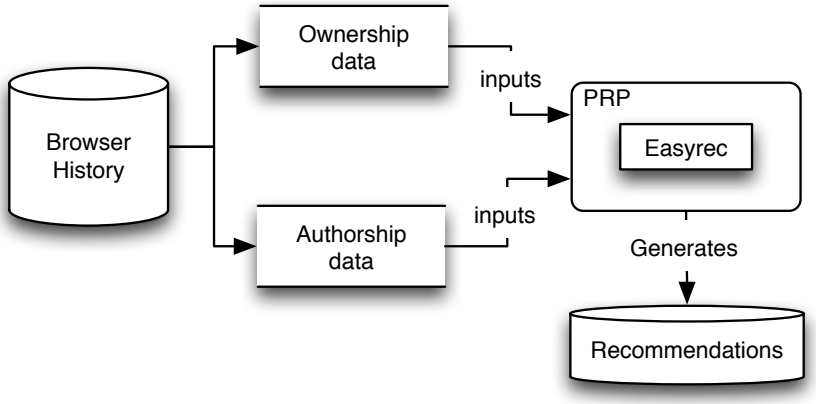

FIGURE 5.6: Experiment set-up without traceability

TABLE 5.5: Easyrec statistics without traceability

Total actions	104875
Total items	104402
Average actions per user	14982
Computed rules	1654

TABLE 5.6: Number of common items without traceability

	User1	User2	User3	User4	User5	User6	User7
User1	-	4	1	13	0	5	8
User2	-	-	35	21	10	7	11
User3	-	-	-	10	22	17	13
User4	-	-	-	-	12	36	24
User5	-	-	-	-	-	14	2
User6	-	-	-	-	-	-	63
User7	-	-	-	-	-	-	-

The number of common ownership and authorship relationships between users is presented in Table 5.6.

The Easyrec Recommender System component of PRP will predict interest of users in resources based in the ownership and authorship information. The prediction results without traceability are presented in Table 5.7.

The recommendation of access privileges by PRP to the resource owner is presented in Table 5.8.

TABLE 5.7: Easyrec predictions without traceability

User	Predictions Number	Useful Predictions	Useful predictions(%)
User1	0	0	-
User2	6	2	33
User3	4	0	0
User4	2	0	0
User5	0	0	-
User6	3	2	66
User7	0	0	-

TABLE 5.8: PRP recommendations without traceability

User	Recommendations	Accepted Recommendations	Accepted Recommendations(%)
User1	1	1	100
User2	2	1	50
User3	0	0	-
User4	7	2	28
User5	0	0	-
User6	3	2	66
User7	1	0	0

TABLE 5.9: Easyrec statistics with traceability

Number of total actions	105005
Number of total items	104402
Average actions per user	15000
Computed rules	20795

5.2.3 Access recommendation with traceability

The access recommendation using traceability information is simulated by considering the resource ownership, authorship and readership data. This set of relations allows simulating collaboration between users. This traceability information is inserted in PRP that will use the Easyrec Recommender System to recommend access privileges to other users as presented in Fig. 5.7.

The Easyrec overal statistics with traceability are presented in Table 5.9.

The number of items that two users have performed actions over them has presented in Table 5.10.

The Easyrec Recommender System component of PRP will predict interest of users in resources based in the traceability information. The prediction results with traceability are presented in Table 5.11.

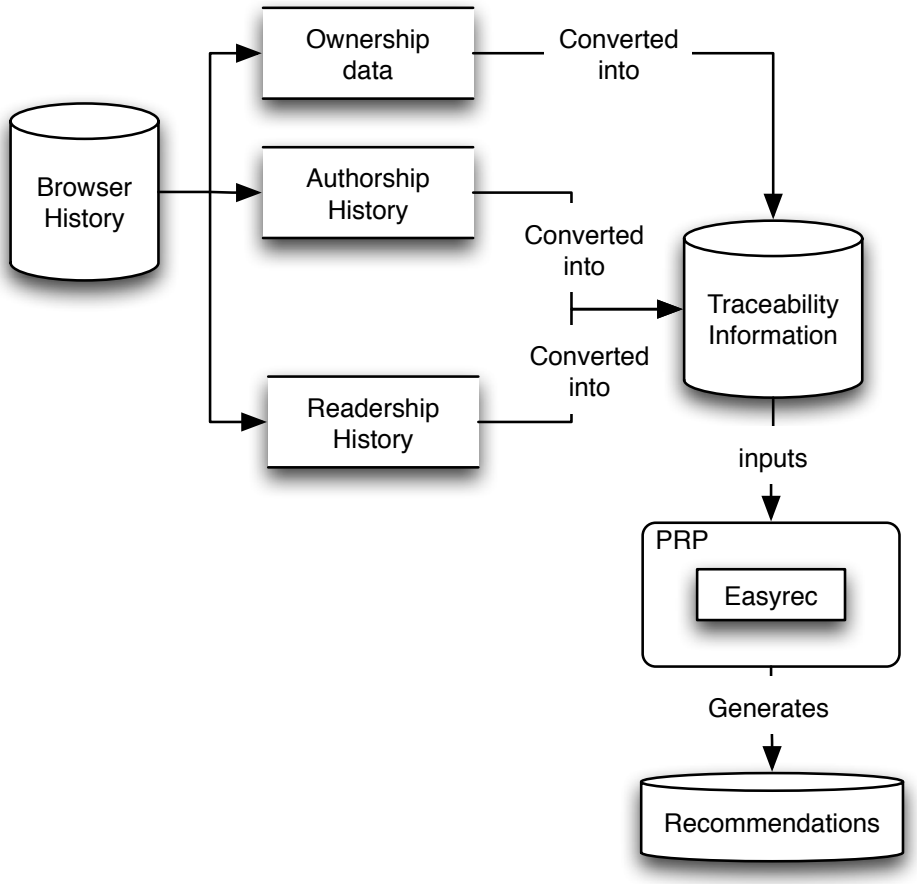

FIGURE 5.7: Experiment set-up with traceability

TABLE 5.10: Number of common items with traceability

	User1	User2	User3	User4	User5	User6	User7
User1	-	19	3	17	8	27	17
User2	-	-	52	65	65	20	22
User3	-	-	-	16	88	35	20
User4	-	-	-	-	55	58	29
User5	-	-	-	-	-	55	29
User6	-	-	-	-	-	-	100
User7	-	-	-	-	-	-	-

TABLE 5.11: Easyrec predictions with traceability

User	Predictions Number	Useful Predictions	Useful predictions(%)
User1	0	0	-
User2	9	6	66
User3	38	29	76
User4	24	12	50
User5	16	6	37
User6	14	4	28
User7	50	24	48

TABLE 5.12: PRP recommendations with traceability

User	Recommendations	Accepted Recommendations	Accepted Recommendations(%)
User1	13	10	77
User2	33	14	42
User3	13	9	69
User4	36	13	36
User5	10	2	20
User6	29	21	72
User7	17	9	53

The recommendation of access privileges by PRP to the resource owner using traceability information and the owner feedback is presented in Table 5.12.

5.2.4 Experiment analysis

Analysis of the experiments will consider both:

- The (average) number of recommendations;
- The usefulness of the recommendations.

The average number of prediction items without traceability is 1,83 items, and 25,1 items with traceability. Based on this figures we can conclude that traceability information improves the average number of total predicted items and for each user as presented in Fig. 5.8.

Based on the survey performed to the users, it was possible to evaluate the usefulness of the recommendations in terms of precision. The average number of useful recommendations per user without traceability is 2,16 items and 21,57 items with traceability. Fig. 5.9 depicts the observations.

The average number of recommendations per user without traceability is 2,16 items where with traceability is 21,57 items. We can conclude that traceability information improves the average number of total recommended items and for each user as presented in Fig. 5.10.

The recommendations to resource owners can be accepted or rejected. There is a big improvement with the utilization of traceability information in the recommendation process as presented in Fig. 5.11.

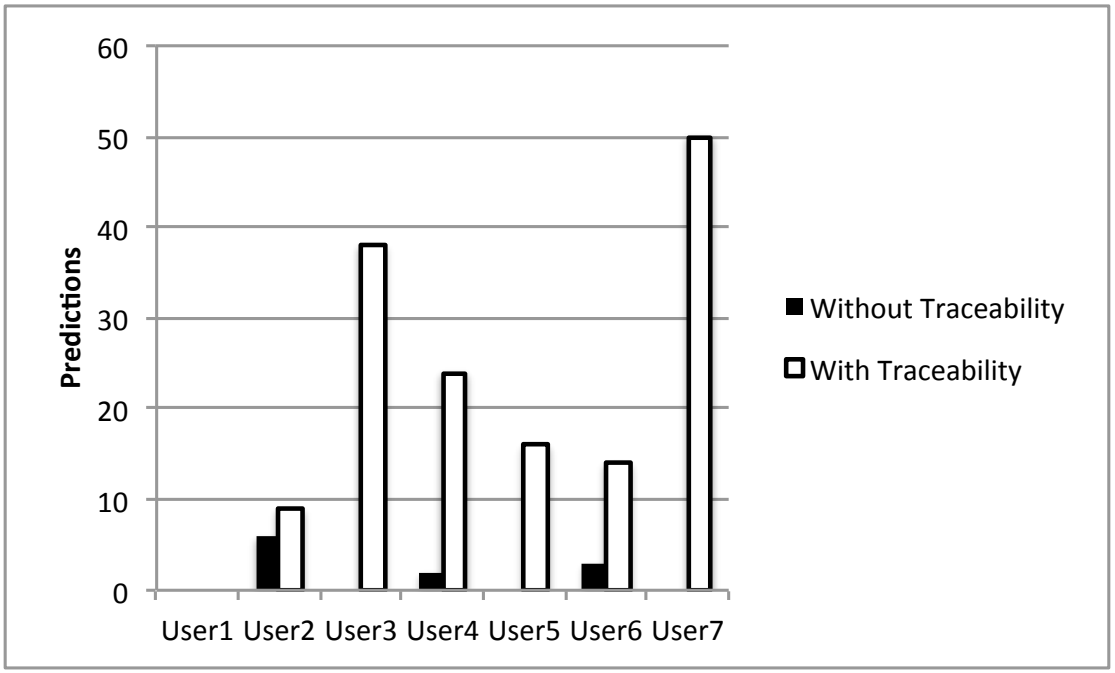

FIGURE 5.8: Predictions with and without traceability

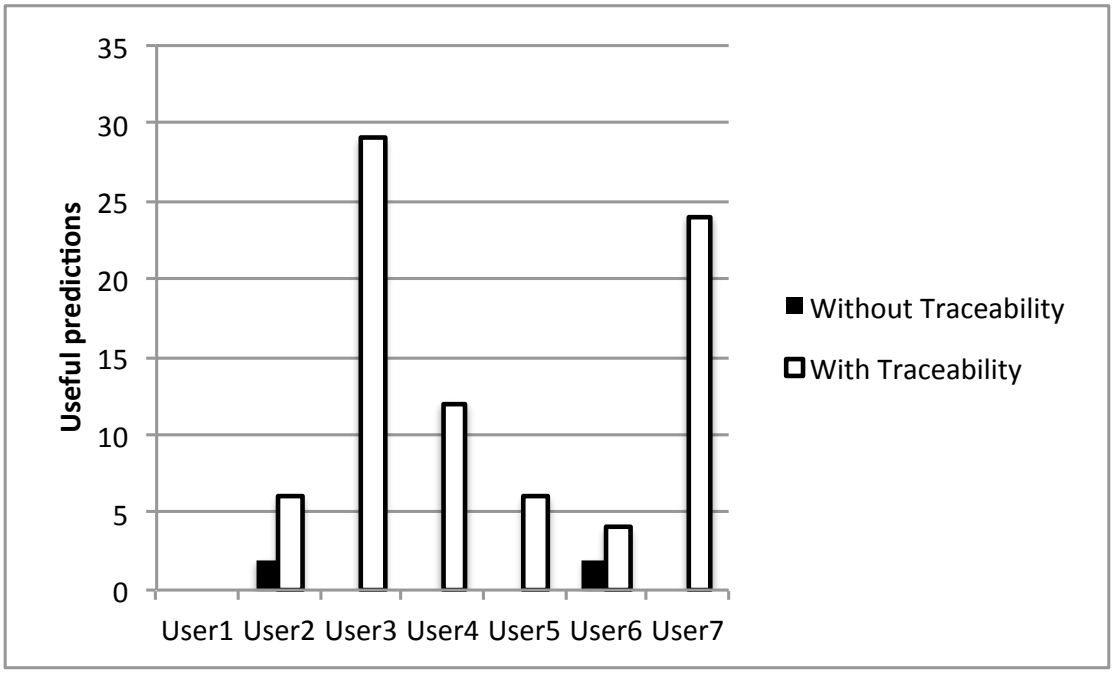

FIGURE 5.9: Useful predictions with and without traceability

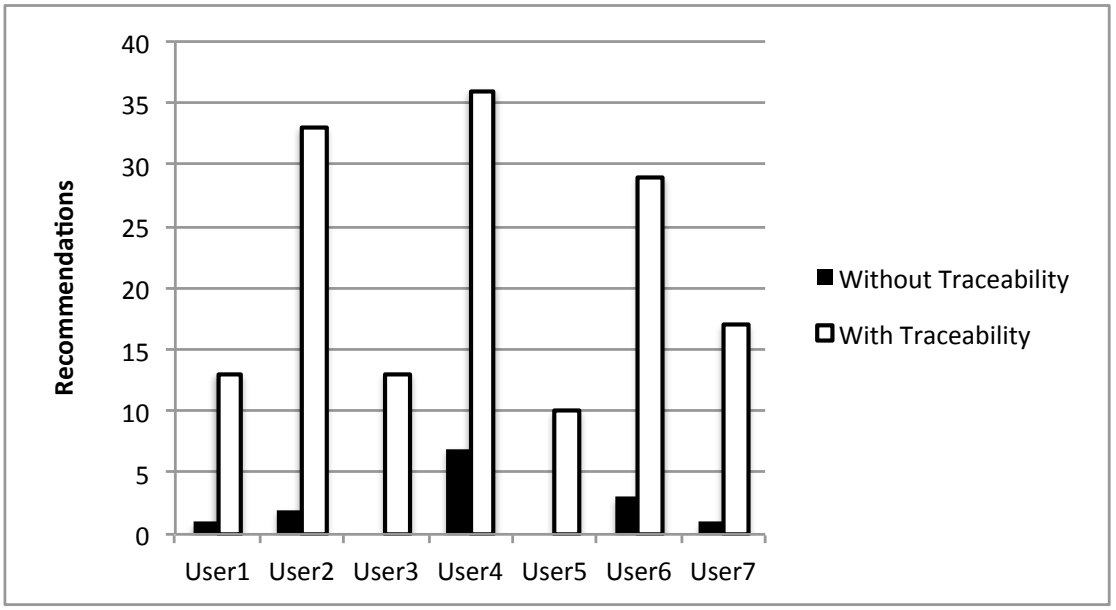

FIGURE 5.10: Recommendations with and without traceability

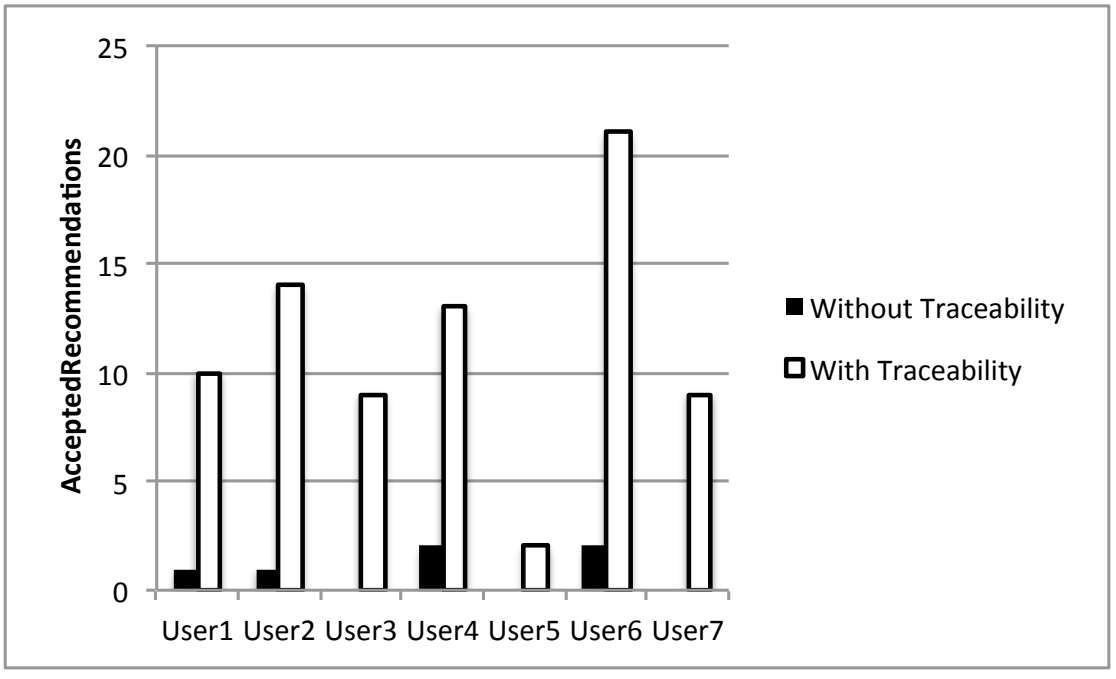

FIGURE 5.11: Accepted recommendations with and without traceability

Chapter 6

Conclusions and Future Work

Recommending access privileges to resources located on different web domains is different (and much harder) than recommending access to resources on the same domain.

In this work it was successfully proposed the enhancement of the previous access control architecture with features capable of predicting and recommending access control privileges.

To achieve the thesis proposal a PRP is introduced in the previous architecture that will recommend access control privileges to resource owners taking advantage of existing traceability information. In order to integrate the PRP into the existing architecture some enhancements were proposed for data modelling over traceability information, feedback information and access privileges. The chosen recommender system (EasyRec) used in that component was not designed for access recommendation purposes hence causing some difficulties in their application. The proposed and developed semantic importer and privileges recommender sub-components can overcome those limitations with success by making semantic interpretations of data. The Privilege Recommender component of the PRP on the other side is capable to interpret and transform the Easyrec predictions into (valid) access privileges recommendations to the owner of the resource. The experiments performed on the system with simulated traceability information manipulated from real users' browsing data is able to demonstrate the effectiveness of the system. The survey conducted among the same users attested that the users largely considered those recommendations suggested by the PRP. In fact, the results obtained can conclude that the traceability information resulting from the users actions is relevant to recommend access privileges by using collaborative techniques, thus enhancing recommendation results. Those results demonstrated that the implemented system is capable of proposing valid access privileges for resources distributed in different domains and therefore improving the quantity and quality of recommendations and allowing users

to gain access to a wider set of resources they might not be aware of. Nevertheless, the process leading to the assessment of coverage of the recommendations was not conclusive as the users were not sufficiently available to answer that part of the survey.

During the thesis development, some ideas emerged. Some are out of the scope of this thesis and others can be used in future work:

- The use of semantic recommendation systems complementing the collaborative recommendation systems;
- The use of content-based recommendation techniques and/or social relationships;
- Develop action sensors to capture the users feedback and help in the access recommendations acceptance or rejection;
- Extend the recommended privileges through ABAC and RBAC concepts;
- Assessing the coverage of recommendations.

Despite all the efforts made in order to provide a useful system evaluation, a real system evaluation would rely on the system being deployed to existing production platforms. Unfortunately, the deployment of such an existing framework arises technical difficulties. In particular, because:

- some of the components must be deployed at server level along with other web server components which raises deployment concerns;
- tracking users' actions on the Internet and keeping them on the cloud might raise some privacy and security issues among users.

Our future efforts are to continue developing this work and possibly deploy it initially in smaller and controlled research environments. In order to overcome the EasyRec cold start problem, some conceptual work relating users and resources to Interests has been depicted in Appendix 2.

It is our belief that deploying this architecture even on smaller scale projects would help in producing enough traceability information and realize what parts of the process could be enhanced in order to overcome some of the depicted issues.

Bibliography

Adomavicius, G., Rokach, L., and Shapira, B. (2011). Recommender Systems Handbook. *Media*, 54:217–253.

Agrawal, R., Imieliński, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. *Proceedings of the 1993 ACM SIGMOD international conference on Management of data SIGMOD 93*, 22(May):207–216.

Airoldi, F., Cremonesi, P., and Turrin, R. (2011). Hybrid algorithms for recommending new items in personal TV. In *2nd Workshop on Future Television at EuroITV 2011 making television personal and social*.

Akrivas, G., Wallace, M., Andreou, G., Stamou, G., and Kollias, S. (2002). Context-sensitive semantic query expansion. In *IEEE international conference on artificial intelligence systems ICAIS*, pages 109–114. IEEE Computer Society.

Bedi, P., Kaur, H., and Marwaha, S. (2007). Trust based recommender system for semantic web. *Search*, pp:2677–2682.

Berkovsky, S., Kuflik, T., and Ricci, F. (2006). Cross-technique mediation of user models. In Wade, V. P., Ashman, H., and Smyth, B., editors, *Adaptive Hypermedia and Adaptive WebBased Systems*, volume 4018 of *Lecture Notes in Computer Science*, pages 21–30. Springer.

Berkovsky, S., Kuflik, T., and Ricci, F. (2007). Mediation of user models for enhanced personalization in recommender systems. *User Modeling and UserAdapted Interaction*, 18(3):245–286.

Berkovsky, S., Kuflik, T., and Ricci, F. (2008). Cross-representation mediation of user models. *User Modeling and UserAdapted Interaction*, 19(1-2):35–63.

Berners-Lee, T. and Cailliau, R. (1990). WorldWideWeb: Proposal for a HyperText Project.

Berners-Lee, T., Fielding, R., and Masinter, L. (2005). Uniform Resource Identifier (URI): Generic Syntax.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The Semantic Web. *Scientific American*, 284(5):34–43.

Bettencourt, N., Peixoto, R., and Silva, N. (2012). Automatic Traceability Acquisition Framework. In *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics - WIMS '12*, page 1, New York, New York, USA. ACM Press.

Bettencourt, N. and Silva, N. (2010). Recommending Access to Web Resources based on User's Profile and Traceability. In *2010 10th IEEE International Conference on Computer and Information Technology*, number Cit, pages 1108–1113. Ieee.

Bizer, C., Heath, T., and Berners-Lee, T. (2009). Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems (IJSWIS)*.

Blair, D. C. (1979). Information Retrieval, 2nd ed. C.J. Van Rijsbergen. London: Butterworths; 1979: 208 pp. Price: 32.50. *Journal of the American Society for Information Science*, 30(6):374–375.

Brickley, D. and Miller, L. (2010). FOAF Vocabulary Specification.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and UserAdapted Interaction*, 12(4):331–370.

Dierks, T. and Allen, C. (1999). RFC 2246: The TLS Protocol.

Fazel-Zarandi, M., Devlin, H. J., Huang, Y., and Contractor, N. (2011). Expert recommendation based on social drivers, social network analysis, and semantic data representation. In *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems HetRec 11*, HetRec '11, pages 41–48. ACM.

Fernández-Tobías, I., Cantador, I., Kaminskas, M., and Ricci, F. (2011). A generic semantic-based framework for cross-domain recommendation. In *Language*, HetRec '11, pages 25–32. ACM.

Finin, T., Joshi, A., Kagal, L., Niu, J., Sandhu, R., Winsborough, W., and Thuraisingham, B. (2008). ROWLBAC: representing role based access control in OWL. SACMAT '08, pages 73–82, Estes Park, CO, USA. ACM.

Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., and Stewart, L. (1999). HTTP Authentication: Basic and Digest Access Authentication.

Gotel, O. and Finkelstein, A. (1994). An analysis of the requirements traceability problem. *Proceedings of IEEE International Conference on Requirements Engineering*, Imperial C(1):94–101.

Gruber, T. R. (1993). A Translation Approach to Portable Ontology Specifications by A Translation Approach to Portable Ontology Specifications. *Knowledge Creation Diffusion Utilization*, 5(April):199–220.

Guha, R., Kumar, R., Raghavan, P., and Tomkins, A. (2004). Propagation of trust and distrust. *Proceedings of the 13th conference on World Wide Web WWW 04*, 133(50):403.

Hartig, O. and Zhao, J. (2012). Provenance Vocabulary Core Ontology Specification.

He, J. and Chu, W. W. (2010). A Social Network-Based Recommender System (SNRS). *Data Mining for Social Network Data*, 12:47–74.

Herlocker, J. L., Konstan, J. A., and Riedl, J. (2000). Explaining collaborative filtering recommendations. *Proceedings of the 2000 ACM conference on Computer supported cooperative work CSCW 00*, pages:241–250.

Housley, R., Polk, W., Ford, W., and Solo, D. (2002). Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile.

Jones, G. J. F. (2005). Challenges and Opportunities of Context-Aware Information Access. *International Workshop on Ubiquitous Data Management*, pages 53–62.

Klenk, A., Heide, T., Radier, B., Salaün, M., and Carle, G. (2009). Pluggable Authorization and Distributed Enforcement with pam_xacml. In *Kommunikation in verteilten Systemen KiVS'09*, pages 253–264.

Konstan, J. A., Mcnee, S. M., Ziegler, C.-n., Torres, R., Kapoor, N., and Riedl, J. T. (2006). Lessons on Applying Automated Recommender Systems to Information-Seeking Tasks. *Artificial Intelligence*, pages 1630–1633.

Kowalczyk, W. and Schut, M. C. (2011). Diversity Measurement of Recommender Systems under Different User Choice Models. *ICWSM*, pages 369–376.

Kuhn, D. R., Coyne, E. J., and Weil, T. R. (2010). Adding Attributes to Role-Based Access Control. *Computer*, 43(6):79–81.

Lebo, T., Sahoo, S., and McGuinness, D. (2012). PROV-O: The PROV Ontology.

Lemire, D. and Maclachlan, A. (2005). Slope One Predictors for Online Rating-Based Collaborative Filtering. *Baseline*, 05(12):471–475.

Linden, G., Smith, B., and York, J. (2003). Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80.

Loizou, A. and Dasmahapatra, S. (2006). Recommender Systems for the Semantic Web. *Design*, 2(4):269–271.

Maamar, Z., Benslimane, D., and Narendra, N. C. (2006). What can context do for web services? *Communications of the ACM*, 49(12):98–103.

Manber, U., Patel, A., and Robison, J. (2000). Experience with personalization of Yahoo! *Communications of the ACM*, 43:35–39.

Moghaddam, S., Jamali, M., Ester, M., and Habibi, J. (2009). FeedbackTrust: using feedback effects in trust-based recommendation systems. *Proceedings of the 2009 ACM RecSys*, pages 269–272.

Moreau, L. and Missier, P. (2012). PROV-DM: The PROV Data Model.

Murakami, T., Mori, K., and Orihara, R. (2008). Metrics for evaluating the serendipity of recommendation lists. *New Frontiers in Artificial Intelligence*, 4914:40–46.

Needleman, M. (2001). RDF: The resource description framework. *Serials Review*, 27(1):58–61.

Noor, S. and Martinez, K. (2009). Using Social Data as Context for Making Recommendations: An Ontology based Approach. *Proceedings of the 1st Workshop on Context Information and OntologiesCIAO 2009 Jun 1 Herakleion Greece*, page #7.

OMahony, M., Hurley, N., Kushmerick, N., and Silvestre (2004). Collaborative recommendation: A robustness analysis. *ACM Transactions on Internet Technology TOIT*, 4(4):344–377.

Pearl, J. (1985). Bayesian Networks: A Model of Self-Activated Memory for Evidential Reasoning. In California, U. O., editor, *Proceedings of the 7th Conference of the Cognitive Science Society*, volume 73, pages 329–334. UCLA Computer Science Department, Computer Science Department, University of California.

Ray, A. R. J. and Kulchenko, P. (2002). Representational State Transfer (REST). *Architectural Styles and the Design of Networkbased*, pages 237–261.

Rescorla, E. (2000). HTTP Over TLS. http://tools.ietf.org/html/rfc2818.

Resnick, P. and Varian, H. R. (1997). Recommender systems. *Communications of the ACM*, 40(3):56–58.

Rucker, J. and Polanco, M. J. (1997). Siteseer: personalized navigation for the Web. *Communications of the ACM*, 40:73–76.

Schafer, J. B., Konstan, J. A., and Riedl, J. (2001). E-Commerce Recommendation Applications. *Data Mining and Knowledge Discovery*, 5(1):115–153.

Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval SIGIR 02*, 46(Sigir):253.

Shardanand, U. and Maes, P. (1995). Social information filtering: algorithms for automating "word of mouth". In Katz, I. R., Mack, R., Marks, L., Rosson, M. B., and Nielsen, J., editors, *Proceedings of the ACM Conference on Human Factors in Computing Systems*, volume 1 of *CHI '95*, pages 210–217. ACM Press/Addison-Wesley Publishing Co., ACM Press/Addison-Wesley Publishing Co.

Shoval, P., Maidel, V., and Shapira, B. (2008). An Ontology-Content-Based Filtering Method. *International Journal on Information Theories and Applications*, 15:303 – 318.

Sieg, A., Mobasher, B., and Burke, R. (2010). Ontology-Based Collaborative Recommendation. *Computing*.

Sinha, R. and Swearingen, K. (1999). Comparing Recommendations Made by Online Systems and Friends. *Interface*.

Story, H., Harbulot, B., Jacobi, I., and Jones, M. (2009). FOAF+SSL: RESTful Authentication for the Social Web. *Current*, pages 1–12.

Studer, R., Benjamins, V. R., and Fensel, D. (1998). Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, 25(1-2):161–197.

Vargas, S. and Castells, P. (2011). Rank and relevance in novelty and diversity metrics for recommender systems. *Proceedings of the fifth ACM conference on Recommender systems RecSys 11*, 107(July):109.

Webster, M. (2012). Provenance - {Merriam}-{Webster} Dictionary. http://www.merriam-webster.com/dictionary/provenance.

Yang, H. and Parthasarathy, S. (2003). On the use of constrained associations for web log mining. *WEBKDD 2002-Mining Web Data for Discovering Usage Patterns and Profiles*, pages 100–118.

Yu, Z., Nakamura, Y., Jang, S., Kajita, S., and Mase, K. (2007). Ontology-Based Semantic Recommendation for Context-Aware E-Learning. In Indulska, J., Ma, J., Yang, L., Ungerer, T., and Cao, J., editors, *Ubiquitous Intelligence and Computing*,

volume 4611 of *Lecture Notes in Computer Science*, pages 898–907. Springer Berlin / Heidelberg.

Zhang, Y., Callan, J., and Minka, T. (2002). Novelty and redundancy detection in adaptive filtering. *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval SIGIR 02*, pp(15):81.

Chapter 7

Appendix 1

7.1 Ontologies

The word "ontology" is used with different meanings in different communities. In the computer science, many definitions of the term ontology exist. The most popular definition is by [Gruber, 1993] who defines an ontology as "an explicit specification of a conceptualization". This definition is further extended to "Ontologies are a formal, explicit specification of a shared conceptualization" by [Studer et al., 1998]. An ontology allows the definition of terms and meanings used to represent areas of knowledge. Ontologies are extremely important in the interaction between systems that constantly exchange knowledge between them. The proper communication of these systems will only be achieved when both systems receive the same interpretation of the implicit knowledge of the documents exchanged. An ontology is generally designed to: (i) enable the use of a semantic knowledge and application; (ii) easy knowledge sharing process between computers and (iii) allow the correct semantic interpretation. To achieve all the previous features, an ontology defines terms and concepts in order to describe and represent specific knowledge domains. The terms and concepts follow a conceptual modelling usually composed by classes, properties and their hierarchical relationships. As conceptual models become more restrict and rules become more complex, the Description Logic (DL) concept gains importance in providing a logical formalism for Ontologies.

Some ontology namespaces and prefixes used in this thesis are presented in the Table 7.1.

TABLE 7.1: Ontology Namespace table

Prefix	Ontology Namespace
foaf	http://xmlns.com/foaf/0.1/
owl	http://www.w3.org/2002/07/owl#
prov	http://www.w3.org/ns/prov#
prv	http://purl.org/net/provenance/ns#
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
xsd	http://xmlns.com/foaf/0.1/

7.2 Semantic Web

The Semantic Web is an extension to the traditional Web in which information has "well-defined meaning, hence better enabling computers and people to work in cooperation" [Berners-Lee et al., 2001]. In the Semantic Web it is possible to associate semantic annotations to resources. The introduction of this type of information allows accurate and precise meaning of information according to an ontology. When the exchange of information was conducted only among humans, with more or less difficulty and with higher capacity by using the conceptual and inference of humans, humans would relate the concepts found in the documents in order to overcome the ambiguity. However, this reality was applicable only when the Internet was intended for human consumption. Currently, with web services automation and the use of intelligent agents, humans and machines have to work with the same information, and it is necessary for applications to be able to interpret the semantic content associated with each document just like a human would. Semantic Web technologies allow us to describe resources using conceptual models, which have clearly defined concepts and their relationships. By using a Semantic Web, systems can understand, for example, the relationship between a person, a place and an event. If a meeting is scheduled for a given place at a given time, the computer can keep this appointment in the person's agenda automatically. Search engines can also benefit from an increase in accuracy allowing their users to anticipate what they are looking for so that the search is restricted not only to keywords, but to the semantics of text.

To reduce the amount of standardisation required and increase reuse, the Semantic Web technologies have been arranged into a model described by several layers as presented in Fig. 7.1.

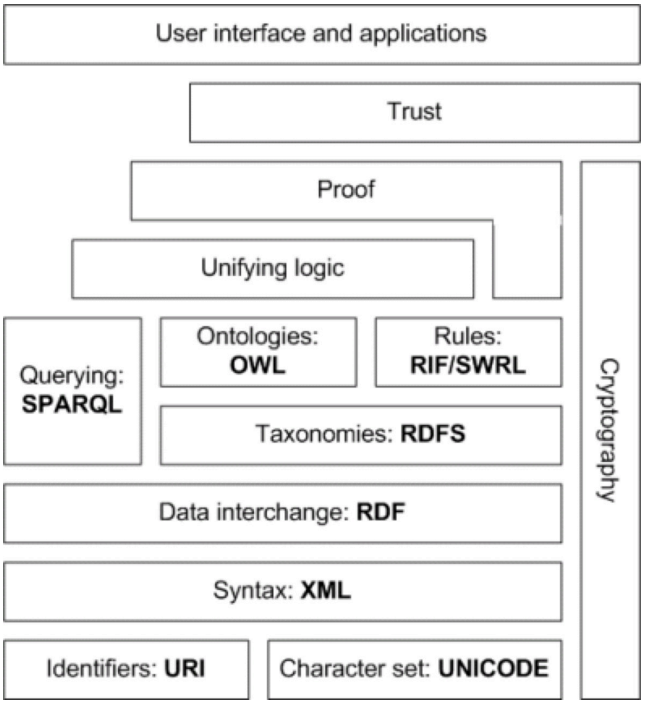

FIGURE 7.1: Semantic Web Stack

Chapter 8

Appendix 2

8.1 Easyrec cold-start

The Easyrec Recommender System works with user actions over resources and relations between the resources. Easryec uses a collaborative filtering technique and as a result suffers from cold start. To improve recommendation accuracy and avoid some of this cold-start, the concept of interests has been created. This approach is based on the principle that items must have one or more associated interests. To distinguish between recommendable items and interests a new type of items called "INTEREST" has been declared on Easyrec for association with interests. In EasyRec only a unique number is required to define a new item and descritpion. In order to define an interest, is used the URI of the ontology concepts that defines interest. This URI is aware of the ontology used but to relate the concepts some previous data processing and modelling must be done. This way the system can give some semantics to the Easyrec recommender system in order to recommend similar items based on interests. In Fig. 8.1 is presented the domain model of this Interest approach.

To create an interest the import-item service of the easyrec REST-API must be used with the item-type value of Interest. To relate the resource with its type it is used an IS-RELATED relation. This relation is obtained from the resources information and is asserted by the PRP semantic importer. One additional action must be created to relate the users and their interests. The action used for this is the Like action, which was deliberately added to the Easyrec system.

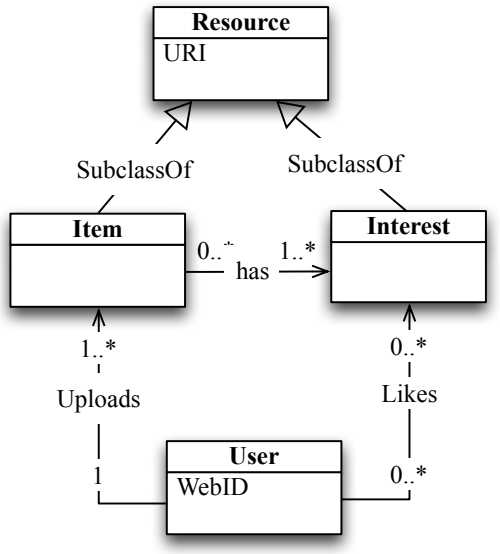

FIGURE 8.1: Easyrec interest domain model

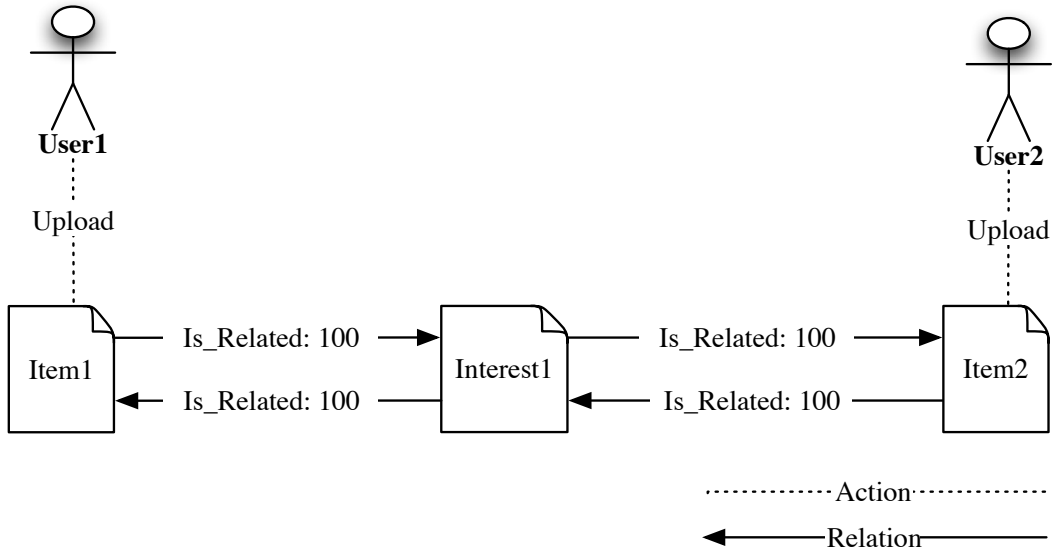

FIGURE 8.2: Conceptual study 1 actions and relations

8.1.1 Conceptual Study 1

In this conceptual study are presented two users (User1, User2) that upload two resources (Item1, Item2) respectively and an Interest (Interest1) as presented in Fig. 8.2. To relate the resource with his type it is used the relation IS-RELATED with the value 100. In this case both Item1 and Item2 share the same Interest (Interest1).

The objective of this conceptual study is to recommend Item1 to User2 and Item2 to User1 with the minimum of possible actions but, only with this actions and relations,

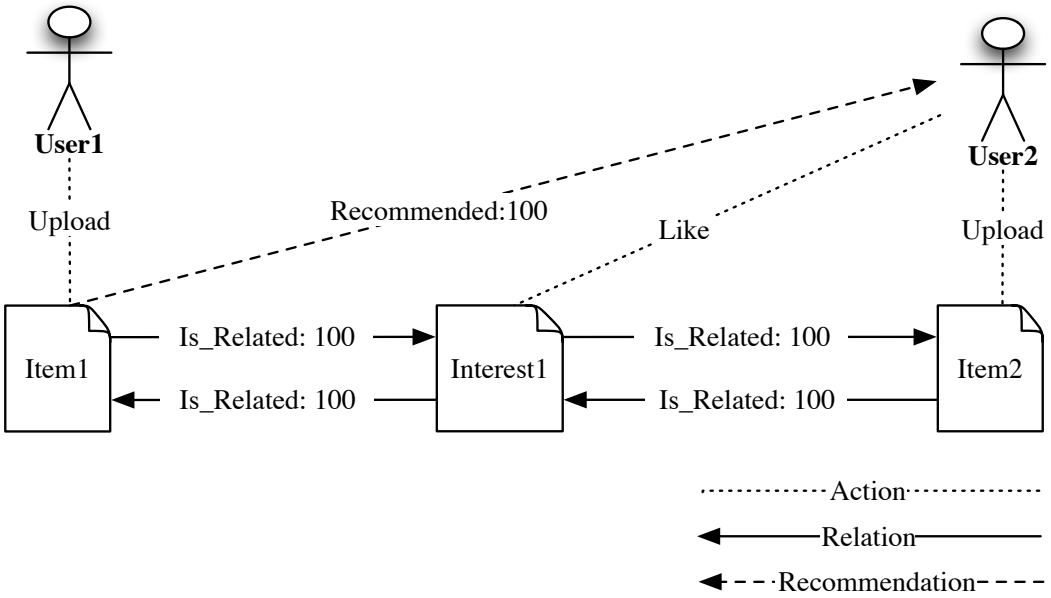

FIGURE 8.3: Conceptual study 1 User2 like actions

TABLE 8.1: Conceptual study 1 User2 recommendations

	Item1	Item2	Interest1
User1	Owned	Not Recommended	Unknown
User2	Recommended	Owned	Like
Interest1	isRelated	isRelated	-

the Easyrec Recommender System cannot recommend these resources because users do not have any direct relation with interests. For evaluation purposes, it is declared that User2 likes Items related to interest Interest1. As presented in Fig. 8.3 the Item1 will be recommended to User2 because the Item1 has the interest Interest1 and User2 has interest in Interest1.

Conceptual study results to User2 are presented by Table 8.1.

As User1 does not have interest in Interest1, the Item2 will not be recommended to User1. In order for Item2 to be recommended to User1, a like action must be added from User1 to the interest Interest1 as in Fig. 8.4.

The recommendation results are presented in the Table 8.2 where the value is the confidence value of the recommendation.

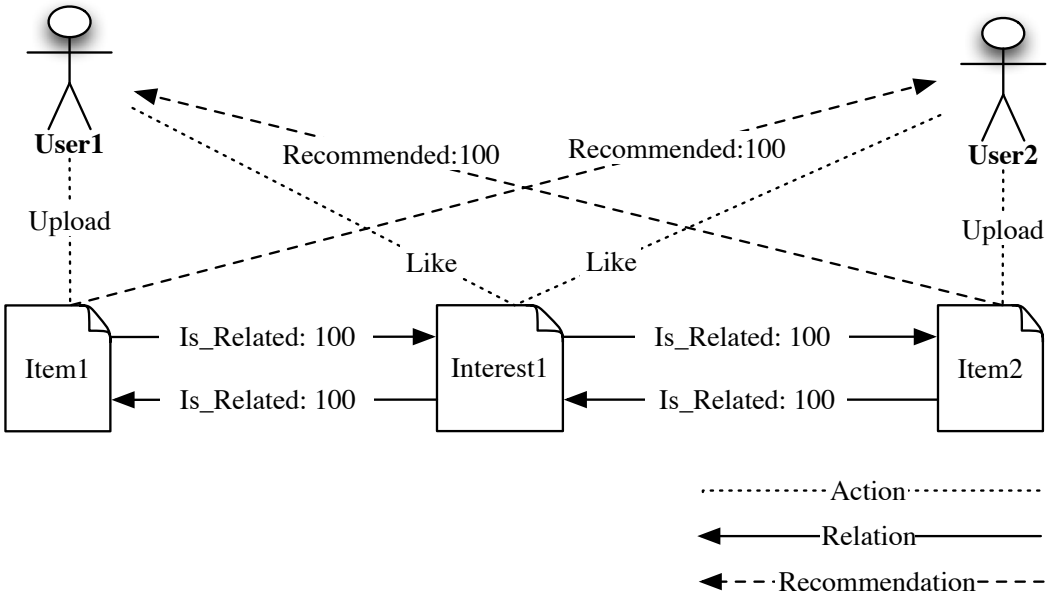

FIGURE 8.4: Conceptual study 1 final actions

TABLE 8.2: Conceptual study 1 final recommendations

	Item1	Item2	Interest1
User1	Owner	Recommended	Like
User2	Recommended	Owner	Like
Interest1	isRelated	isRelated	-

8.1.2 Conceptual Study 2

In conceptual study 2 three items have been uploaded (Item1, Item2, Item3), by three users (User1, User2, User3), respectively. There are also two interests (Interest1 and Interest2). As presented in Fig. 8.5, Item1 and Item 2 are related to Interest1, and Item2 and Item3 are related with Interest2. Only with these actions, none of these resources will be recommended.

In order for recommendation of items to take place, User1 and User2 are assigned to like Interest2 as presented in Fig. 8.6.

The result is the recommendation of Item2 and Item3 to User 1 and the recommendation of Item2 to User3. The final result of this conceptual study is presented in Table 8.3.

8.1.3 Conceptual Study 3

In the previous conceptual studies there is an assumption that all users have at least one action over resources. In this conceptual study we only have two items (Item1, Item2)

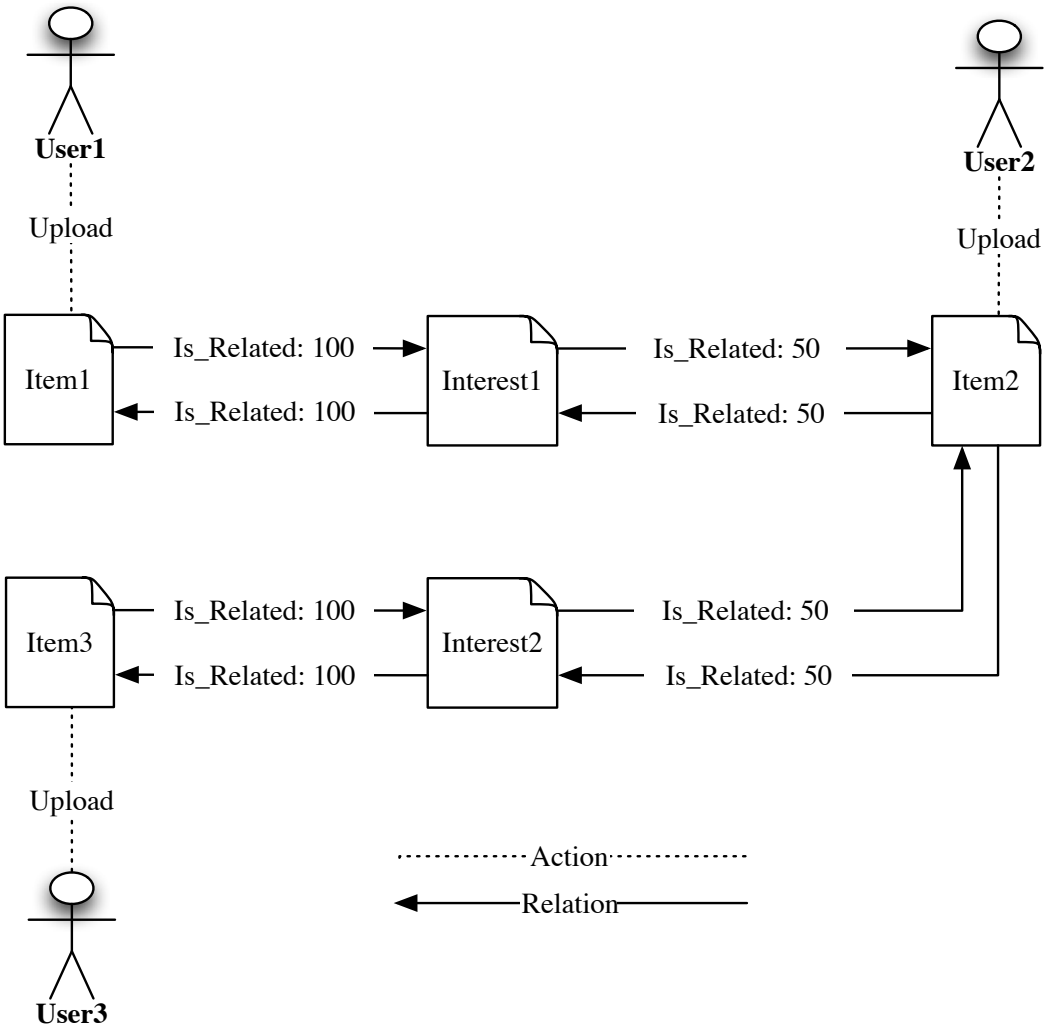

FIGURE 8.5: Conceptual study 2 actions and relations

TABLE 8.3: Conceptual study 2 recommendations

	Item1	Item2	Item3	Interest1	Interest2
User1	Owner	Recommended	Recommended	Unknown	Like
User2	Not Recommended	Owner	Not Recommended	Unknown	Unknown
User3	Not Recommended	Recommended	Owner	Unknown	Like
Interest1	isRelated	isRelated	-	-	-
Interest2	-	isRelated	isRelated	-	-

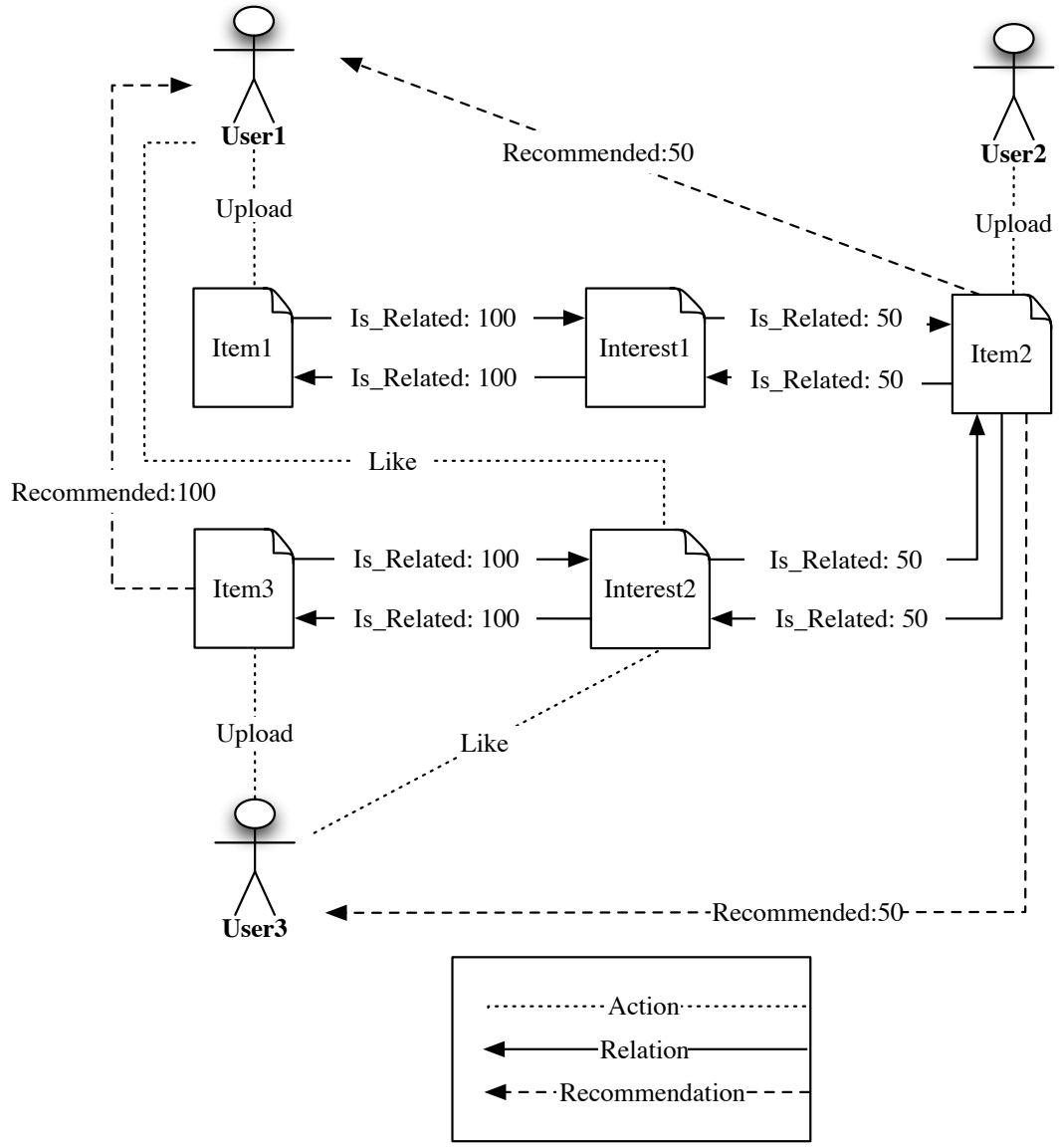

FIGURE 8.6: Conceptual study 2 recommendations

TABLE 8.4: Conceptual study 3 recommendations

	Item1	Item2	Interest1
User1	Recommended	Recommended	Like
Interest1	isRelated	isRelated	-

associated with the interest Interest1 and a single user User1 that only has the interest Interest1 without performing any action over any resource as presented in Fig. 8.7.

The recommended resources in this case will be Item1 and Item2 with the confidence value of 100. The final result of this conceptual study is presented in Table 8.4.

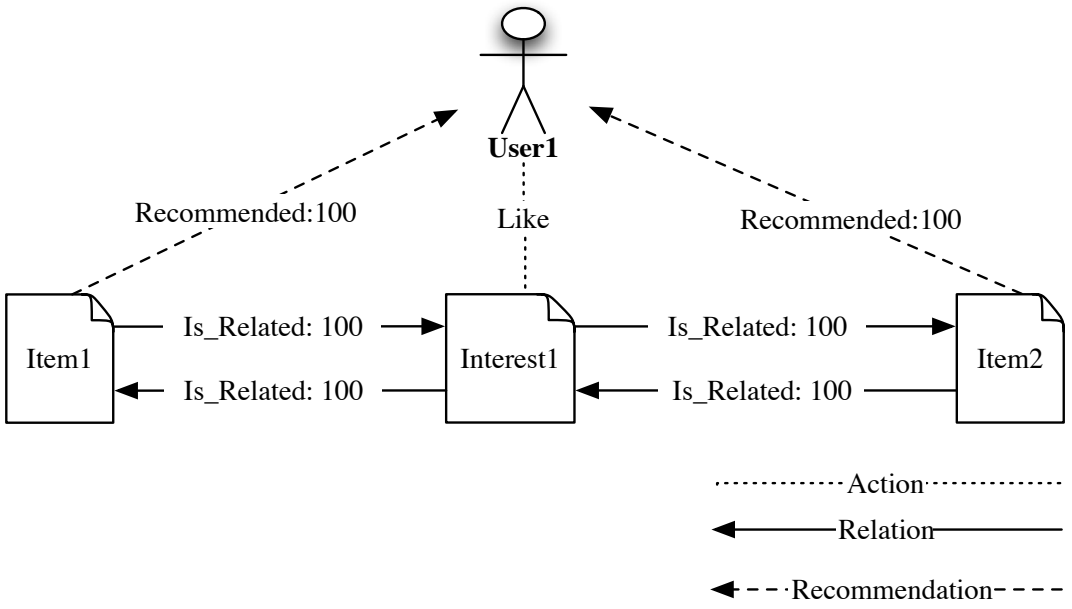

FIGURE 8.7: Conceptual study 3 actions and recommendations

8.2 Test Case

In order to validate the PRP, experiments with real users have been done. The users must select interests, upload resources to a platform and associate them to the available interests. First, users must select their interests. After that, each user must submit at least 3 resources. For each uploaded item, at least one interest must be assigned to the resource in order to avoid some of the Easyrec collaborative cold start. The resource upload and interest association compose the resource submission. The Resource submission process is presented in Fig. 8.8.

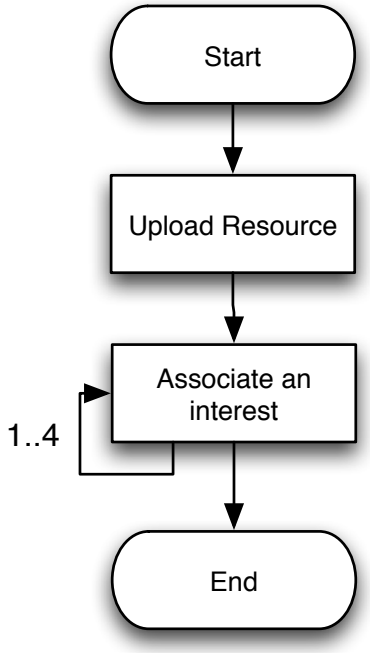

FIGURE 8.8: Resource Submission