



Vrije Universiteit Brussel

FACULTY OF ENGINEERING

Transforming 2D Online Communities to 3D Virtual World: The FOAF and SIOC Based 3D Approach

John Mark Ssebunnya

Promoter: Prof. dr. Olga De Troyer

Co-Promoter(s): Dr. Bram Pellens

Dr. Frederic Kleinermann

Thesis submitted in partial fulfilment of the requirements for the Master of Applied
Sciences & Engineering: Applied Computer Science

Academic year 2009-2010



Abstract

This thesis explores transforming 2D online communities to content-rich 3D Virtual Worlds. Generating content-rich 3D applications from 2D online communities is faced with challenges of extraction and integration of the content. These challenges are currently addressed with the OpenID and OpenSocial initiatives. However, we believe that OpenID is limited to sharing profile information not data like photos, videos, articles etc whereas OpenSocial suffered teething problems due to privacy and security concerns. We thus explore how semantic approaches such as “Friend of a Friend” (FOAF) and “Semantically interlinked online communities” (SIOC) can be used to address this problem of extraction and integration of the content hence enable us to gather data from online communities.

Virtual Worlds bring users closer to a real world view, interaction and manipulation of objects. Virtual Worlds are becoming more popular on the web given the technological changes that we have seen in the past few years with new lighter file formats that make Virtual Worlds better accessible through the Web. As a result of this development we are increasingly seeing virtual versions of some social networks, virtual shopping centers, virtual gaming services and other Virtual World experiences. We argue that existing 3D Virtual Worlds such as ExitReality and Second Life are limited in content since they are focused on getting individual pages or individual communities and representing them in 3D. Our aim is to be able to capture content from different online communities and represent this data in a 3D Virtual World.

Our thesis makes an important contribution by developing and testing the “FOAF and SIOC based 3D Approach”. With it, we are able to prove that with open standards such as FOAF and SIOC, it is possible to move data between online communities. We also demonstrate that the approach to gather data from different online communities and store it in a technology independent format can be a starting point of a big Mashup. In order to ease the development of complex Virtual Worlds, we propose extensions to CoDePA and VR-WISE with its OntoWorld tool. The extension on CoDePA is an addition of static patterns for complex static objects. The proposed extension to OntoWorld will use XSL Template as the back end at the mapping phase. In addition, a front end extension to the OntoWorld graphical editor is proposed to ease selection and customization of the static patterns.

Acknowledgements

Two years Masters study has been an exciting, enriching and challenging period. This thesis is a testimony that my efforts and the efforts of those who have guided and helped me have been worthwhile. Without the help, support, and encouragement from several persons, I would never have been able to make it to this point.

First and foremost, I would like to express sincere gratitude to my promoter, Prof. Dr. Olga De Troyer, Head of the Web & Information Systems Engineering (WISE) Laboratory of the Vrije Universiteit Brussel for having considered and accepted my request to do this thesis in the WISE Lab and wisely allocated me the best resources for guidance along the way.

I am deeply grateful to Dr. Bram Pellens and Dr. Frederic Kleinermann, my co-promoters, for the countless effort they have made to help me understand the concepts, to guide and proof read this thesis. Many thanks for giving me the first real exposure to professional academic writing.

I extend my appreciation to all my professors during the 2yrs of my Masters study. I am completing my Master Program in Engineering: Applied Computer Science with a great wealth of knowledge courtesy of your great service.

Having arrived at Vrije Universiteit Brussel with little or no knowledge of what good and bad to expect, I sincerely appreciate Mrs. Irene Raadschelders for having been exceptionally receptive and incredibly helpful at every stage of my study, providing all the information and advice I needed.

My family, relatives and friends have been a big pillar in all my academic progress and life success. I take this opportunity to thank you for supporting me, believing and trusting in me even in the hardest of times. I would never live without you!

Lastly, my deep and sincere gratitude to the Vrije Universiteit Brussel as well as the International Relations and Mobility Office for having offered me a scholarship for my Masters study. I believe this thesis and my entire academic achievement is proof that the scholarship was well seeded. Without it I would never have done this Masters program. Many Thanks.

Contents

Abstract	ii
Acknowledgements	iii
Glossary	ix
1 Introduction	1
1.1 2D and 3D Online Communities	1
1.2 Semantics for Data Portability.....	3
1.3 Motivation	5
1.4 Thesis Aims and Structure	5
1.4.1 Aims	5
1.4.2 Structure.....	6
2 Background and Related Work	7
2.1 Web 2.0 Online Communities.....	7
2.2 Semantic Web: Web 3.0	8
2.3 Ontologies.....	10
2.3.1 FOAF	13
2.3.2 SIOC	15
2.3.3 FOAF and SIOC in use	18
2.3.3.1 MyExperiment.....	19
2.3.3.2 Semantic Social Network	20
2.4 Facebook Open Graph	20
2.5 XML and X3D.....	21
2.6 XSL Transformations	26
2.7 Virtual World	28
2.7.1 Second Life	28
2.7.2 Vivaty.....	29
2.7.3 ExitReality	32
2.8 Website Transformation Approaches.....	35
2.8.1 Dante	35
2.8.2 SADle	36
3 FOAF and SIOC Based 3D Approach	39
3.1 Social Media Mining	41
3.1.1 FOAF Mining	41

3.1.2	SIOC Mining	45
3.1.3	Integration	47
3.2	3D Transformation.....	48
3.2.1	Template Mapping	48
3.2.2	Code Generation	49
3.3	Adopting the Approach	50
4	Case Study.....	51
4.1	Preparatory Steps	51
4.1.1	Set up of Online Communities	51
4.1.2	Creation of FOAF Profile	52
4.2	Using the Approach	54
4.2.1	Phase 1: Social Media Mining.....	54
4.2.1.1	Provide FOAF Profile	54
4.2.1.2	FOAF Mining: Extract FOAF Data	54
4.2.1.3	SIOC Mining: Extract SIOC Data	57
4.2.1.4	Integration: Person Communities Ontology	60
4.2.2	Phase 2: 3D Transformation.....	60
4.2.2.1	Template Mapping: Choose XSL Templates.....	60
4.2.2.2	Code Generation: Transform to 3D Environment.....	61
5	Integration with Existing Approaches.....	63
5.1	VR-WISE Approach.....	63
5.1.1	Specification Stage	63
5.1.2	Mapping Stage.....	64
5.1.3	Generation Stage.....	65
5.2	CoDePA	65
5.2.1	Specifying Patterns	67
5.2.1.1	Pattern Front-End Specification.....	67
5.2.1.2	Pattern Back-End Specification	67
5.2.1.3	Pattern Library	67
5.2.2	Using Patterns	68
5.3	Integration	68
5.3.1	Templates Generation.....	68
5.3.1.1	Specification Phase	69
5.3.1.2	Mapping Phase.....	70
5.3.1.3	Generation Phase.....	71

5.3.2	Linking with VR-WISE.....	71
5.3.3	Extending CoDePA and VR-WISE.....	71
6	Conclusion and Future Work.....	75
6.1	Summary.....	75
6.2	Contributions and Achievements.....	75
6.3	Limitations and Future Work.....	76
	Bibliography.....	78
	Appendix A: PHP Snippet.....	83

Table of Figures

Figure 1-1 : ExitReality Sample 3D Space Look	3
Figure 1-2: Many Accounts on Online Communities	4
Figure 2-1 : Popular Social Networks Services Features	8
Figure 2-2 : Semantic Web Framework.....	10
Figure 2-3 : Ontology Graphical representation	12
Figure 2-4 : FOAF Classes.....	14
Figure 2-5 : FOAF RDF.....	15
Figure 2-6 : SIOC core ontology model.....	16
Figure 2-7 : Modeling social data with SIOC	17
Figure 2-8 : SIOC FOAF Relation	18
Figure 2-9 : MyExperiment Ontology Modules Architecture.....	20
Figure 2-10 : X3D architecture	23
Figure 2-11 : X3D Hello World	25
Figure 2-12 : X3D Hello World in Browser.....	25
Figure 2-13 : XSLT Transformation	26
Figure 2-14 : XSLT Example.....	27
Figure 2-15 : Second Life Insight	29
Figure 2-16 : Vivaty Studio	30
Figure 2-17 : Vivaty Player.....	31
Figure 2-18 : Box X3D	31
Figure 2-19 : ExitReality Facebook Space.....	33
Figure 2-20 : ExitReality WannerBros.com.....	33
Figure 2-21 : ExitReality Selecting Themes.....	34
Figure 2-22 : ExitReality CNN.com.....	34
Figure 2-23: Dante.....	36
Figure 2-24 : SADle	37
Figure 3-1 : Transformation Stages	40
Figure 3-2 : Start FOAF Extract	42
Figure 3-3 : FOAF Data Structure.....	43
Figure 3-4 : Friends List	44
Figure 3-5 : FOAF Mining.....	44

Figure 3-6 : SIOC Mining	45
Figure 3-7: Sample Exported RDF	46
Figure 3-8: Integration Files	48
Figure 3-9: Main XSL File Skelton	49
Figure 4-1 : WordPress with SIOC RDF data under Semantic Radar	52
Figure 4-2 : Created FOAF Profile Extract.....	53
Figure 4-3 : FOAF Link Form	54
Figure 4-4 : PHP and XSL Snippet to Extract FOAF Profile.....	55
Figure 4-5 : Preview FOAF Data.....	56
Figure 4-6 : Extracted FOAF Data	57
Figure 4-7 : List User's 2D Online Communities.....	58
Figure 4-8 : Sample SIOC Site RDF	59
Figure 4-9 : Sample Post RDF Extract	60
Figure 4-10 : Concepts Mapped to Primitive Objects	61
Figure 4-11 : Choosing Templates	61
Figure 4-12 : Sample XSLT Template	62
Figure 5-1 : VR-WISE Approach	64
Figure 5-2 : VR-WISE Approach - Virtual Shop	65
Figure 5-3 : CoDePA Pattern Specification Framework.....	66
Figure 5-4: Domain Specification Extract	69
Figure 5-5: World Specification Extract.....	70
Figure 5-6 : OntoWorld Integration.....	73

Glossary

API	Application Programming Interfaces
CoDePA	Conceptual Design Pattern Approach
DOM	Document Object Model
FOAF	Friend of a Friend
GGG	Giant Global Graph
OAI-ORE	Open Archives Initiative Object Reuse and Exchange
OWL	Web Ontology Language
RDF	Resource Description Framework
RDFa	Resource Description Framework – in – attributes
RDFS	RDF Schema
SADie	Structural-Semantics for Accessibility and Device Independence
SIOC	Semantically Interlinked Online Communities
VE	Virtual Environments
VR	Virtual Reality
VRML	Virtual Reality Modelling Language
VR-WISE	Virtual Reality – With Intuitive Specifications Enabled
VW	Virtual Worlds
X3D	Extensible 3D
XSLT	Extensible Style sheet Language Transformations

1 Introduction

We are living in the era of Web 2.0 and a promising time when new applications involving the use of Virtual Worlds and/or the emerging Semantic Web are being felt. Web 2.0 is a set of economic, social and technology trends that collectively form the basis for the next generation of the Internet—a more mature, distinctive medium characterized by user participation, openness and network effects (O’Reilly, et al., 2006). Though “openness” is apparently a characteristic of Web 2.0, we still see a lot of online communities¹ where sharing of data is done within the confines of that online community and thus challenges of data portability are inherent in Web 2.0. Data portability challenges in addition to technical complexity of 3D Virtual World development make the realization of rich content 3D Virtual Worlds harder. The Semantic Web that derives from Sir Tim Berners-Lee's vision of the Web as a universal medium for data, information and knowledge exchange has resulted in development of ontologies such as “Friend of a Friend” (FOAF) and “Semantically Interlinked Online Communities” (SIOC) which can be exploited to address the challenges of data portability. In this thesis, we study how to use FOAF and SIOC ontologies to retrieve data from different 2D online communities and we develop and test a method to transform this data to 3D Virtual Worlds.

1.1 2D and 3D Online Communities

The brands Facebook, MySpace, Wikipedia and Blogger are among some of the top Internet brands with some of the highest online traffic as rated by web traffic reporting services such as Alexa² and comScore³. Unlike brands like Coca Cola, Shell, which have been established for quite so many years, these Internet brands are averagely about 8 years old as of 2010 but the rate at which they became popular has been interesting to watch e.g Facebook alone has 400 million users⁴. This is courtesy of Web 2.0 that has brought about online communities where ‘freedom of speech’ is largely encouraged and a high level of interaction with Rich Internet Application technologies can be used. Jo Kim defines an online community as a virtual community that exists online whose members enable its existence through taking

¹ <http://news.bbc.co.uk/2/hi/technology/6944653.stm>

² <http://www.alexa.com/topsites>

³ <http://www.comscore.com/>

⁴ <http://www.facebook.com/press/info.php?statistics>

part in membership rituals. In online communities such as bulletin boards, anyone can post content whereas in weblogs only a restricted number of people can initiate posts. In some cases like wikis such as Wikipedia, pages can be created and edited by many users. Social networks such as Facebook, MySpace, Twitter, and YouTube are contributing to a larger percentage of today's current traffic. A social network is a social structure made of individuals (or organizations) called "nodes" which are tied (connected) by one or more specific types of interdependency, such as friendship, kinship, financial exchange, dislike, sexual relationships, or relationships of beliefs, knowledge or prestige (Safko, et al., 2009).

Although the Internet bandwidth has increased and reaching more people every day, most of the Web 2.0 online communities are still only available in 2D format. Second Life⁵ and ExitReality⁶ are some of the few popular 3D Virtual Worlds. ExitReality promises to transform the social networking experience by offering virtual versions of every social network site profile. However ExitReality and other 3D Virtual Worlds are still very limited in what content they represent since they cannot yet incorporate content from different 2D online communities. Figure 1-1 represents a sample look into ExitReality 3D Space. It shows an avatar of a person and her MySpace 3D space.

Virtual Environments (VEs) or Virtual Worlds (VWs) refer to the computer-generated (and computer-maintained) three-dimensional environments, which can be navigated and interacted with by the users and are often the result of a Virtual Reality system (Pellens, 2007). Virtual Reality is about creating acceptable substitutes for real objects or environments, and it is not really about constructing imaginary worlds that are indistinguishable from the real world (Vince, 2004). Thus, in 3D online communities we expect to see 3D objects as acceptable substitutes for real world objects. The avatar of a person in Figure 1-1 is an illustration of this concept.

⁵ <http://www.secondlife.com>

⁶ <http://www.exitreality.com/>



Figure 1-1 : ExitReality Sample 3D Space Look⁷

1.2 Semantics for Data Portability

The boom of online communities has come with its challenges, among which are the massive data and many profiles users have to maintain. It is common for one user to hold accounts with Facebook, MySpace, Wikipedia, Hi5, Orkut, Bebo, on several forums, blogs and many other communities. Additionally, the credentials required are different across the social networks. For example, while some online community may require a username other prefers e-mail addresses as identifiers. Figure 1-2 shows a sample of several communities on which one can have accounts. Having a uniform username is not easy since preferred usernames are often unavailable. Therefore, users end up having many usernames for these services. In the long run users risk forgetting some and thus losing time to recover these details when they visit these services. Projects like OpenID⁸ have come to address some of these problems (Rehman, 2008). On their home page OpendID.net* has a message with these words “Can’t remember your passwords? Tired of filling out registration forms? OpenID is a safe, faster,

⁷ <http://virtualunderworld.net/wordpress/?p=114>

⁸ <http://openid.net/>

and easier way to log in to web sites.” OpenID authentication is used and provided by several large websites. Providers include AOL, BBC, Google, IBM, Microsoft, MySpace, Orange, PayPal, VeriSign, LiveJournal, Yandex, Ustream and Yahoo⁹. However, there are still challenges to overcome among which, OpenID is limited to sharing profile information not data like photos, videos, articles etc. Thus, users will still have data spread across online communities. We, consequently, see a behavior like that of an information silo¹⁰ and a problem of data portability. OpenSocial¹¹ is another initiative that would have addressed this in the circles of social networks (Lynne, 2009). Google¹² states that OpenSocial is a set of common application programming interfaces (APIs) for web-based social network applications, developed by Google along with MySpace and a number of other social networks. However, it suffered teething problems due to privacy and security concerns.



Figure 1-2: Many Accounts on Online Communities

In this era of Web 3.0, popularly known as the semantic web, the use of technologies such as Resource Description Framework (RDF) to add meaning to web pages is starting to be adopted. At the top conceptual level, the use of ontologies to represent concepts is also

⁹ <http://www-03.ibm.com/press/us/en/pressrelease/23461.wss>
¹⁰ An information silo is a management system incapable of reciprocal operation with other, related management systems.
¹¹ <http://code.google.com/apis/opensocial/>
¹² <http://www.google.com/intl/en/press/pressrel/opensocial.html>

being highly researched and starting to be deployed. One of the most cited definitions of Ontology is by Gruber, “ontology is a formal, explicit specification of a shared conceptualization” (Gruber, 1995). The FOAF and SIOC ontologies are semantic approaches with which we can address the problem of data portability. When the challenges of data portability come into the equation, realization of 3D Virtual Worlds with content from different online communities is taken to another level of complexity.

1.3 Motivation

Although 2D online communities present many data usage options, their level of interaction is limited and not so close to the experience of the real world. Virtual Worlds bring users closer to a real world view, interaction and manipulation of objects. Despite the enhanced interaction options Virtual Worlds provide, generation of 3D Virtual Worlds is a cumbersome task that requires a lot of technical expertise and time. This is taken to another level of complexity with the challenges of data portability discussed earlier. We thus need to find ways that will reduce the level of technical expertise, time and cost required to extract data from different communities and transform this data to 3D Virtual World.

The growing use of semantic web driven by the desire to enable computers to understand the meaning of information on the web is a step in the right direction introducing ontologies such as FOAF and SIOC which are yet to be fully tested and deployed. Thus, it is important to study the extent to which they can be deployed to generate solutions for extracting data from different communities and transforming this data to 3D Virtual World. This will be the focus of this thesis.

1.4 Thesis Aims and Structure

We will now provide the aims of this thesis before describing how the thesis is structured.

1.4.1 Aims

1. Develop an approach to capture content from different online communities.
2. Store captured content into an XML format that can potentially be a start of a big mashup.
3. Create an approach to transform this data into 3D Virtual Worlds.
4. Develop the necessary tool that will support this approach i.e. the three steps above.

5. Build a 3D Virtual World case study using the captured data from different online communities.

1.4.2 Structure

- Chapter 1 has introduced online communities and the semantics for data portability in the context of this work as well as the motivation and aims of the thesis.
- Chapter 2 provides background information on Online Communities, Semantic Web, Ontologies such as FOAF and SIOC, Virtual World and XSLT Transformation. The chapter also covers related work such as SecondLife.com and insights into the implications of new social trends such as Facebook Open Graph.
- Chapter 3 describes our approach: the “FOAF and SIOC Based 3D Approach”. In it we detail how the approach works to gather data from online communities and transform this data to 3D Virtual World.
- Chapter 4 presents a case study transforming several 2D online communities into 3D Virtual World. In this case study, we demonstrate how our online FOAF and SIOC Based 3D Approach support tool is used.
- Chapter 5 discusses the relation and integration with existing approaches such as the VR-WISE approach and CoDePA. This includes how to use VR-WISE to generate 3D templates. We also propose extensions to CoDePA and VR-WISE.
- Chapter 6 elaborates on contributions, future work and gives the conclusion of this thesis.

2 Background and Related Work

In Chapter 1 we introduced online communities and semantics for data portability under which we looked at Web 2.0 and the Semantic Web. In this chapter, we discuss Web 2.0 online communities in more detail to understand their building blocks. This discussion is followed by a look at the semantic web (Web 3.0) in which we discuss ontologies in more depth. This gives us a background to discuss FOAF and SIOC which are the two core ontologies that we use to enable exportation of data in a machine readable format. We also discuss the adoption of FOAF and SIOC and insights into the implications of new social trends such as Facebook Open Graph. We follow this with a look at XML, X3D and XSLT. X3D is the XML based language that we shall use to represent online communities in 3D and XSLT will enable us to do a transformation from our ontology to X3D. We explore related Virtual World projects such as SecondLife.com, Vivaty and ExitReality to understand current 3D online communities. We end this section with a look at website transformation approaches such as Dante and SADle which give us an insight into how transformation of website content from one format into another has been approached previously.

2.1 Web 2.0 Online Communities

The main features on which online communities are based are users and content. Users are the generators of content. Content can be in forms such as photos, music, text and videos. Tagging has been popularized by Web 2.0 enabling users of online communities to define tags as some form of metadata used to describe content such as articles, photos, videos etc. The way content is managed, the specialties in content and the specialties in users are some of the factors with which classification of these communities is drawn. Most common communities are Social Networks, Blogs, Wikis and Forums. Taking a look at social networks, we realize that most, if not all, have the ability to create and share a personal profile which typically includes a photo, some basic personal information such as name, age, sex, location. Furthermore attributes like favorite bands, books, TV shows, movies, hobbies and Web sites are also added. Most social networks on the Internet also let you post photos, music, videos and personal blogs on your profile page. However, the most important feature of online social networks is the ability to find and make friends with other site members. Social networking sites typically have a section dedicated to comments by friends (Safko, et al., 2009).

A comparison we made of two very popular social networks Facebook and MySpace shows that they have common features such as

- Member Personal Profiles
- Friends lists
- Photos
- Comments by friends

Their service is broken down and branded as depicted in Figure 2-1.

Facebook		MySpace
Publisher	Video	Bulletins
News Feed	Chat	Groups
Wall	Public Profiles	MySpaceIM
Photos	Easter Eggs	MySpaceTV
Notes	UserNames	Applications
Gifts	Platform applications	MySpace Mobile
Marketplace	APIs	MySpace News
Pokes	Facebook Markup Language	MySpace Classifieds
Status	Facebook Connect	MySpace Karaoke
Events		MySpace Polls
Networks and groups		MySpace forums

Figure 2-1 : Popular Social Networks Services Features

Looking at the services for the two social networks, it is evident that their strength is in the way they are using content. As the saying goes “Content is King”, this user generated content combined with the interactive technologies of Web 2.0 has been a cornerstone in the boom of social networks and other online communities such as Wikipedia (The Wisdom of Crowds).

2.2 Semantic Web: Web 3.0

The Web 2.0 communities discussed earlier have largely been built at the time when semantics were not yet fully used over the web, though this is changing fast. We envision a time when, our computers can scan our friends’ posts over the web and perfectly give us results of the posts that are related to the topic that we are interested in. This will happen with the semantic web. Berners-Lee argues that the Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation. The Semantic Web will enable machines to comprehend semantic documents and data, not human speech and writings

(Berners-Lee, et al., 2001). According to the World Wide Web Consortium (W3C), the Semantic Web is a vision: the idea of having data on the Web defined and linked in a way that it can be used by machines not just for display purposes, but for automation, integration and reuse of data across various applications.

Two important technologies for developing the Semantic Web are already in place: eXtensible Markup Language (XML) and the Resource Description Framework (RDF). XML lets everyone create their own tags—hidden labels that annotate entire Web pages or sections of text on a page. Scripts, or programs, can make use of these tags in sophisticated ways. However, the script writer has to know what the page writer uses each tag for. In short, XML allows users to add arbitrary structures to their documents but says nothing about what the structures mean. Meaning is expressed by RDF, which encodes it in sets of triples, each triple being rather like the subject, verb and object of an elementary sentence.

Figure 2-2 represents the different components that together make up the framework of the semantic web by offering the following purposes among others

- URI/IRI for a unique identification of semantic web resources.
- Unicode for representation of text in different languages.
- XML for the interchange of structured data over the Web, XML Namespaces to integrate (qualify) markup from multiple sources, XML Query (XQuery) to query collections of XML data and XML Schema to define structure (grammar) of specific XML languages.
- RDF Model & Syntax define RDF triples and represent resource information in a graph plus describe taxonomies based on RDF Schema (RDFS).
- Ontology OWL is the language to define vocabularies that extends RDFS with more advanced features (e.g. cardinality) and enables reasoning based on description logic.

In the following section, we shall explore the Ontology layer.

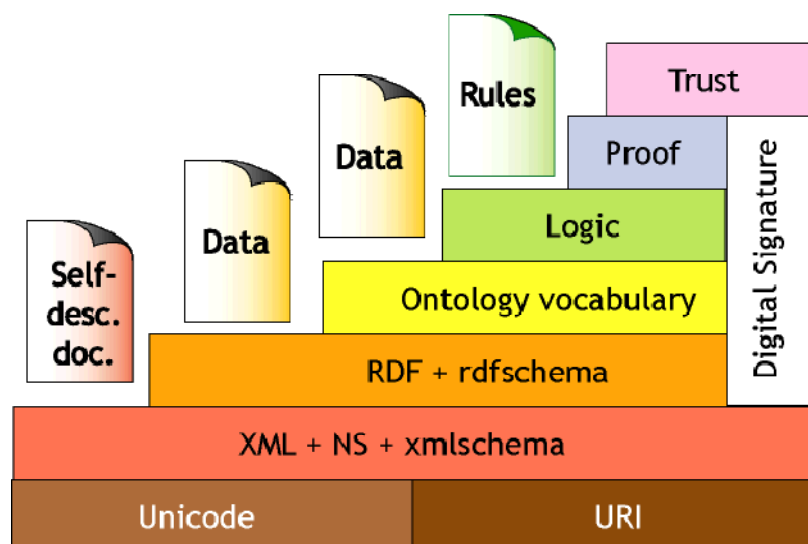


Figure 2-2 : Semantic Web Framework¹³

2.3 Ontologies

When told to go to the bank, it can be debated if you are told to go to the commercial institution, to a river bank or some other place they call a bank. This ambiguity and arising disagreement on terms happens quite often even in computing. The main hurdle to agent communications is that different agents adopt different repositories of terms, and even if they adopt the same terminology, it cannot be assumed that the terms will mean the same thing. To solve this ambiguity, the notion of ontology has been adapted. The notion of ontology originates from the discipline of Philosophy. It has evolved to its current meaning in the context of Computer and Information Science where it refers to a designed artefact which formally represents agreed semantics in a computer resource. This enables the sharing and reuse of information and allows for the interoperation of information resources (Guarino, et al., 1995).

As stated earlier one of the most cited definitions proposed by Gruber, "Ontology is a formal, explicit specification of a shared conceptualization (Gruber, 1995)." Fensel analyzes this definition identifying four main concepts involved: an abstract model of a phenomenon termed "conceptualization", a precise mathematical description hints the word "formal", the precision of concepts and their relationships clearly defined are expressed by the term "explicit" and the existence of an agreement between ontology users is hinted by the term "shared" (Fensel, 2003).

¹³ <http://www.semantic-conference.com/primer.html>

From these definitions, we can identify some essential aspects of ontologies:

- Ontologies are used to describe a specific domain.
- The terms and relations are clearly defined in that domain.
- There is a mechanism to organize the terms (commonly a hierarchical structure is used as well as IS-A or HAS-A relationships).
- There is an agreement between users of ontology in such a way that meaning of the terms is used consistently.

Semantics may be implicit, existing only in the minds of the humans who communicate and build Web applications. They may also be explicit and informal, or they may be formal. The further we move along the continuum, the less ambiguity there is and the more likely it is to have robust correctly functioning Web applications. For implicit and informal semantics, there is no alternative to hardwiring the semantics into Web application software. In the case of formal semantics, hardwiring remains an option, in which case the formal semantics serve the important role in reducing ambiguity in specifying Web application behaviour, compared to implicit or informal semantics. There is also the new possibility of using automated inference to process the semantics at runtime. This would allow for much more robust Web applications, in which agents automatically learn something about the meaning of terms at runtime (Uschold, 2003).

Building ontology is a similar process to building an object oriented program. However, classes and objects in a program are about data structures, whereas classes and objects in ontologies are about the domain (Farquhar, et al., 1997). Another approach to build ontology is proposed by (Gómez, 1998). It follows the next set of principles:

- Clarity and objectivity: Ontology should provide precise and objective definitions in natural language form.
- Completeness: Definitions should be expressed by necessary and sufficient conditions.
- Coherence: Ontology should allow making consistent inferences with definitions.
- Maximal monotonic extendibility: Inclusion of new general or specific terms should be included in the ontology without revision of existing definitions.
- Minimal ontological commitment: Few axioms as possible about the modelled world.
- Ontological Distinction Principle: Classes with different identity criteria must be disjoint.

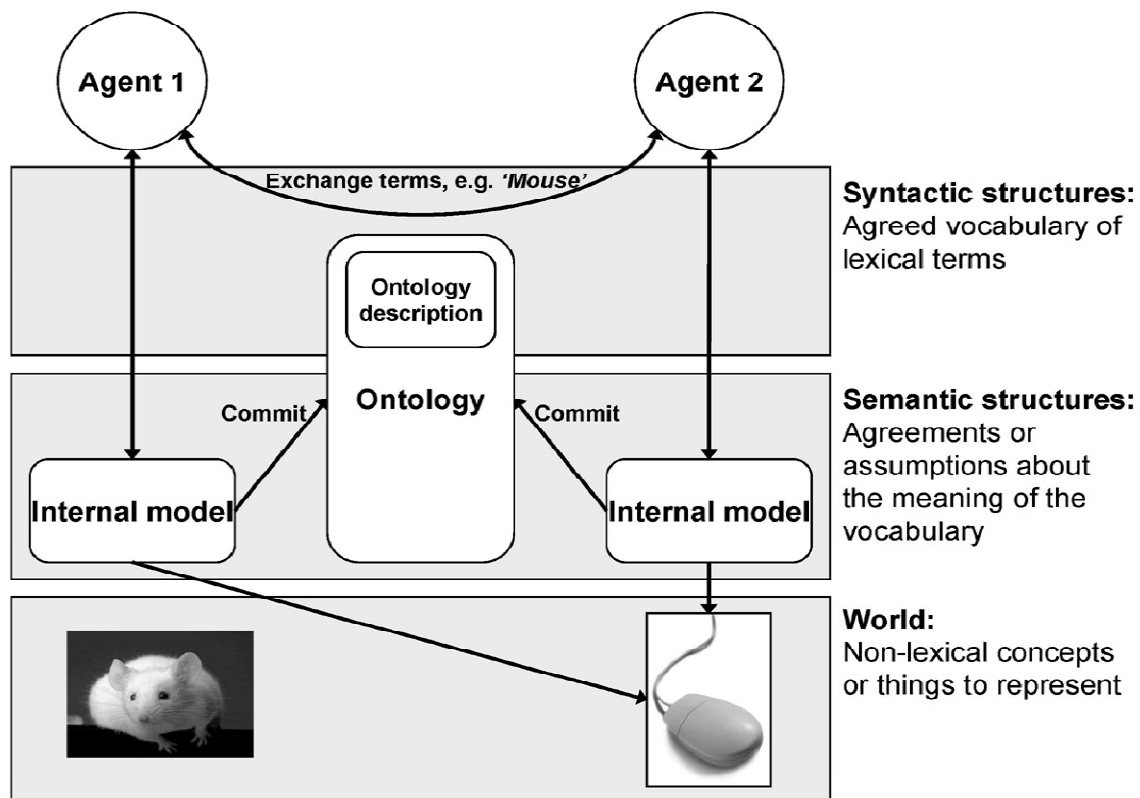


Figure 2-3 : Ontology Graphical representation

The moment we accept that agents may use the same terms to mean different things, we need a way for an agent to discover what another agent means when it communicates. In order for this to happen, every agent will need to publicly declare the terms in use and their meaning. This specification is commonly referred to as the agent's ontology (Gruber, 1995). If it was written only for people to understand; this specification could be just a glossary. However, meaning must be accessible to other software agents. This requires that the meaning be encoded in some kind of formal language. This will enable a given agent to use automated reasoning to accurately determine the meaning of other agent's terms. For example as seen in Figure 2-3, suppose Agent 1 sends a message to Agent 2 and in this message is a pointer to Agent 1's ontology. Agent 2 can then look in Agent 1's ontology to see what the terms mean, the message is successfully communicated, and the agent's task is successfully performed. At least this is the theory. In practice there is a plethora of difficulties. The main objective is for this to happen consistently, reliably, and fully automatically. Most of these difficulties arise from various sources of heterogeneity. For example, there are many different ontology representation languages, different modeling styles and inconsistent use of terminology, to name a few (Uschold, 2003).

2.3.1 FOAF

FOAF¹⁴ is a machine-readable ontology describing persons, their activities and their relations to other people and objects. FOAF allows the expression of personal information and relationships, and is a useful building block for creating information systems that support online communities thus taking the social networking aspect of the Web one step further (Golbec, et al., 2008). The FOAF vocabulary, originated by Dan Brickley and Libby Miller, gives a basic expression for community membership: describing people and their basic properties such as name, e-mail address, and so on. FOAF is expressed using the Resource Description Framework (RDF) and the Web Ontology Language (OWL). Anyone can use FOAF to describe him or herself. Computers may use these FOAF profiles to find, for example, all people living in Europe, or to list all people both you and a friend of yours know¹⁵. This is accomplished by defining relationships between people. Each profile has a unique identifier (such as the person's e-mail addresses, a Jabber ID, or a URI of the homepage or weblog of the person), which is used when defining these relationships.

Tim Berners-Lee in a 2007 essay redefined the Semantic web concept into something he calls the Giant Global Graph¹⁶ (GGG), where relationships transcend networks/documents. He considers the GGG to be on equal grounds with Internet and World Wide Web, stating that "I express my network in a FOAF file, and that is a start of the revolution (Segaran, et al., 2009)."

The FOAF Vocabulary Specification¹⁷ is a dictionary of named properties and classes. Objects such as "Person", "Online Account" are defined with classes and attribute such as "family name", "first name" are defined with properties. A person begins by describing himself or herself using the foaf:Person class, listing key identity attributes such as name, gender, and resources relating to them. They can also list their interests, and each person is uniquely identified by using the foaf:mbox property containing their email address (it is still useful to consider how this is normally the primary key for online 'signup' forms). An alternative identification property is foaf:openid conforming to the OpenID perspective of using a unique single URI to establish the identity. The person in question then moves on to

¹⁴ <http://www.foaf-project.org/>

¹⁵ <http://www.ibm.com/developerworks/xml/library/x-foaf.html>

¹⁶ <http://dig.csail.mit.edu/breadcrumbs/node/215>

¹⁷ <http://xmlns.com/foaf/spec/>

describing their friends, each friend is an instance of the foaf:Person class (Rowe, et al., 2008).

Figure 2-4 shows some of the classes with attributes of the FOAF ontology and the relationships between them. On it we can read that the foaf:Person called “Mathew Rowe” holds a foaf:OnlineAccount on <http://www.facebook.com> and “Mathew Rowe” knows another foaf:Person who also holds a foaf:OnlineAccount on <http://www.facebook.com>

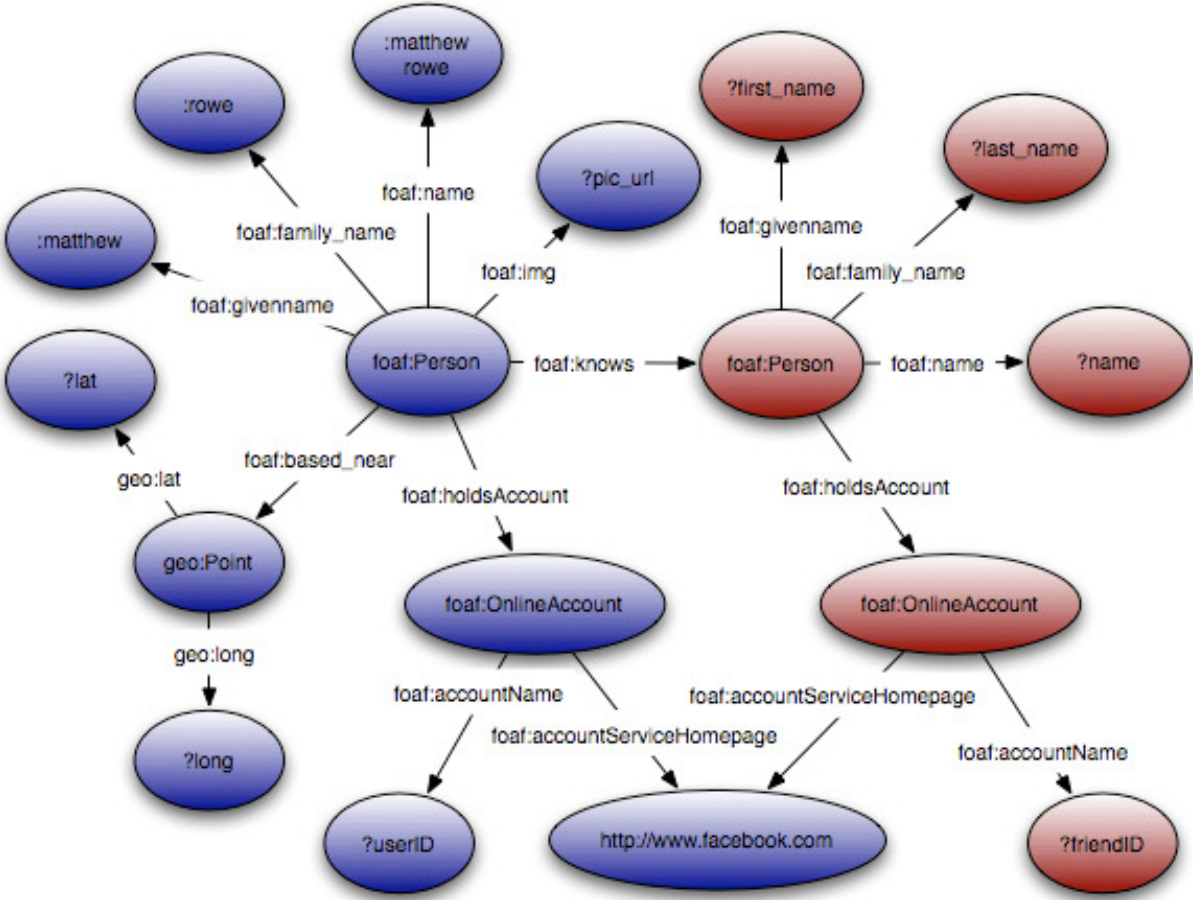


Figure 2-4 : FOAF Classes¹⁸

A simple FOAF RDF document to represent the concepts in Figure 2-4 would look like the extract in Figure 2-5

¹⁸ <http://www.dcs.shef.ac.uk/~mrowe/images/foafv1.jpg>

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:admin="http://webns.net/mvcb/">
<foaf:PersonalProfileDocument rdf:about="">
</foaf:PersonalProfileDocument>
<foaf:Person rdf:ID="me">
<foaf:name>Mathew Rowe</foaf:name>
<foaf:givenname>Mathew</foaf:givenname>
<foaf:family_name>Rowe</foaf:family_name>
<foaf:nick>Mathew Rowe</foaf:nick>
<foaf:mbox_sha1sum>b6f15f6fc345c395d95282ae780391d3d12e44f8</foaf:mbox_sha1sum>
<foaf:depiction rdf:resource="pic_url"/>
<foaf:knows>
<foaf:Person>
<foaf:name>Mark</foaf:name>
<foaf:mbox_sha1sum>4c8eb3253c3ba31388e6d899238530488544c6db</foaf:mbox_sha1sum>
<rdfs:seeAlso
rdf:resource="http://www.facebook.com"/></foaf:Person></foaf:knows></foaf:Person>
</rdf:RDF>

```

Figure 2-5 : FOAF RDF

Creating a FOAF file can be done easily on FOAF-a-Matic¹⁹, a form based tool that will generate a basic description for you that will cover the most commonly used FOAF data items. In addition some social networking sites such as LiveJournal, Tribe.net, recently Facebook and more²⁰ can export FOAF format data using user created applications.

2.3.2 SIOC

SIOC²¹ is a Semantic Web technology that provides methods for interconnecting online communities such as blogs, forums and mailing lists to each other. It consists of the SIOC ontology, an open-standard machine readable format for expressing the information contained both explicitly and implicitly in the online communities. Figure 2-6 (W3C, 2008) shows the six main classes of the SIOC core ontology²² namely: Site, Forum, Post, Usergroup, User and Role. These abstract class definitions offer a flexible range for the capture of knowledge existing in online communities. The concept sioc:User can be thought of as the central point for the ontology, sioc:User is a sub class of the foaf:OnlineAccount class. The SIOC ontology also defines properties that relate each of the classes to the user in a similar manner to how a user interacts within an online community (Rowe, et al., 2008).

¹⁹ <http://www.ldodds.com/foaf/foaf-a-matic>

²⁰ <http://wiki.foaf-project.org/w/DataSources>

²¹ <http://sioc-project.org/>

²² <http://rdfs.org/sioc/ns>

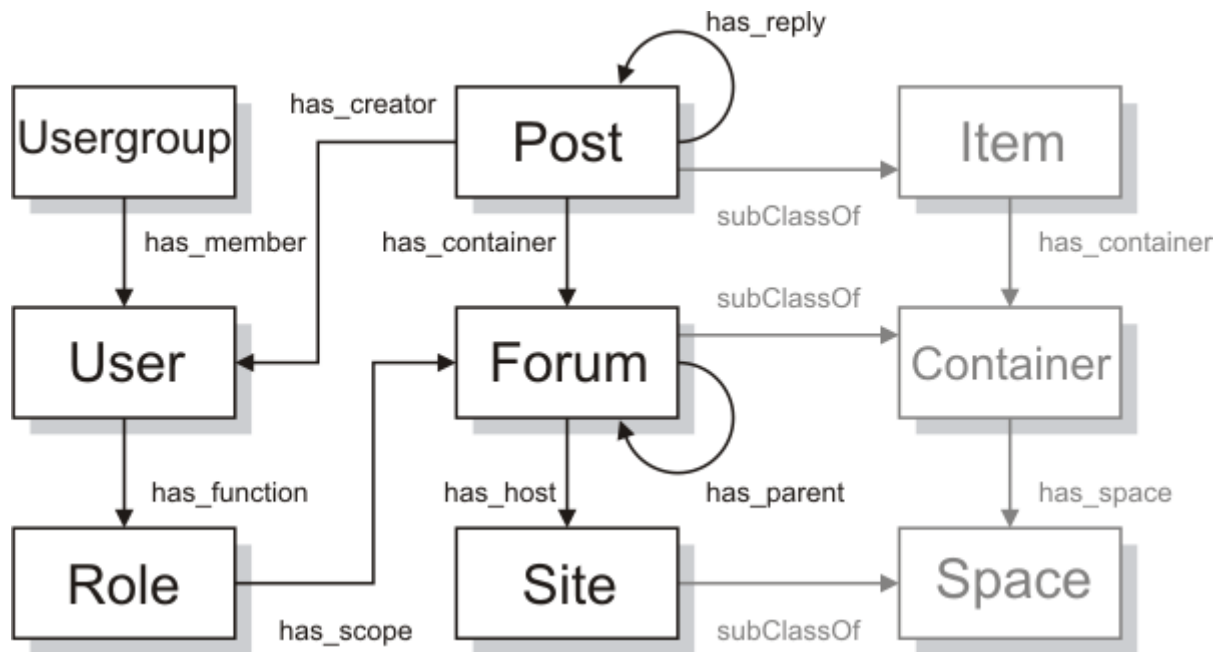


Figure 2-6 : SIOC core ontology model²³

The SIOC vocabulary is based on RDF and is defined using RDFS. SIOC documents may use other existing ontologies to enrich the information described. Additional information about the creator of the post can be described using the FOAF Vocabulary and the foaf:maker property. Rich content of the post (e.g., an HTML representation) can be described using AtomOWL or RSS 1.0 Content module.

While the SIOC core ontology provides a sufficient model to describe the activities of online communities, it is not fine-grained enough to allow one to distinguish between, e.g., a blog post and an update on a microblogging service, as it only introduces a single sioc:Post class. In order to define this additional level of semantics, a SIOC Types module was designed, modeling more than 20 different types of content, such as sioc:MicroblogPost and sioc:WikiArticle.

SIOC reuses the Dublin Core²⁴ model to define various attributes of created content (such as the creation date of an item, using dcterms:created) and FOAF to model personal identity and related attributes. Interlinking FOAF and SIOC also provides a model for identity federation on the Web. While one can have different user profiles on different websites (represented as instances of sioc:User), they all relate to the same physical person

²³ <http://www.w3.org/2008/09/msnws/papers/sioc.html#f2>

²⁴ <http://dublincore.org/>

(foaf:Person). Figure 2-7 shows, a model for interoperability and portability of social data between services. It has 3 layers namely “People”, “Services and User Accounts” and “Container and Content Items”. In it we can read that a physical person “foaf:Person” called “Bob” holds a sioc:User account “Roberto” on “delicious” and a sioc:User account “robertr” on “wordpress.com”. Each of these sioc:User accounts links to Container and Content Items such as sioc:Post that the user has made on the services. In this case we see a post like “Happy New Year” which “robertr” has made on his Wordpress.com blog.

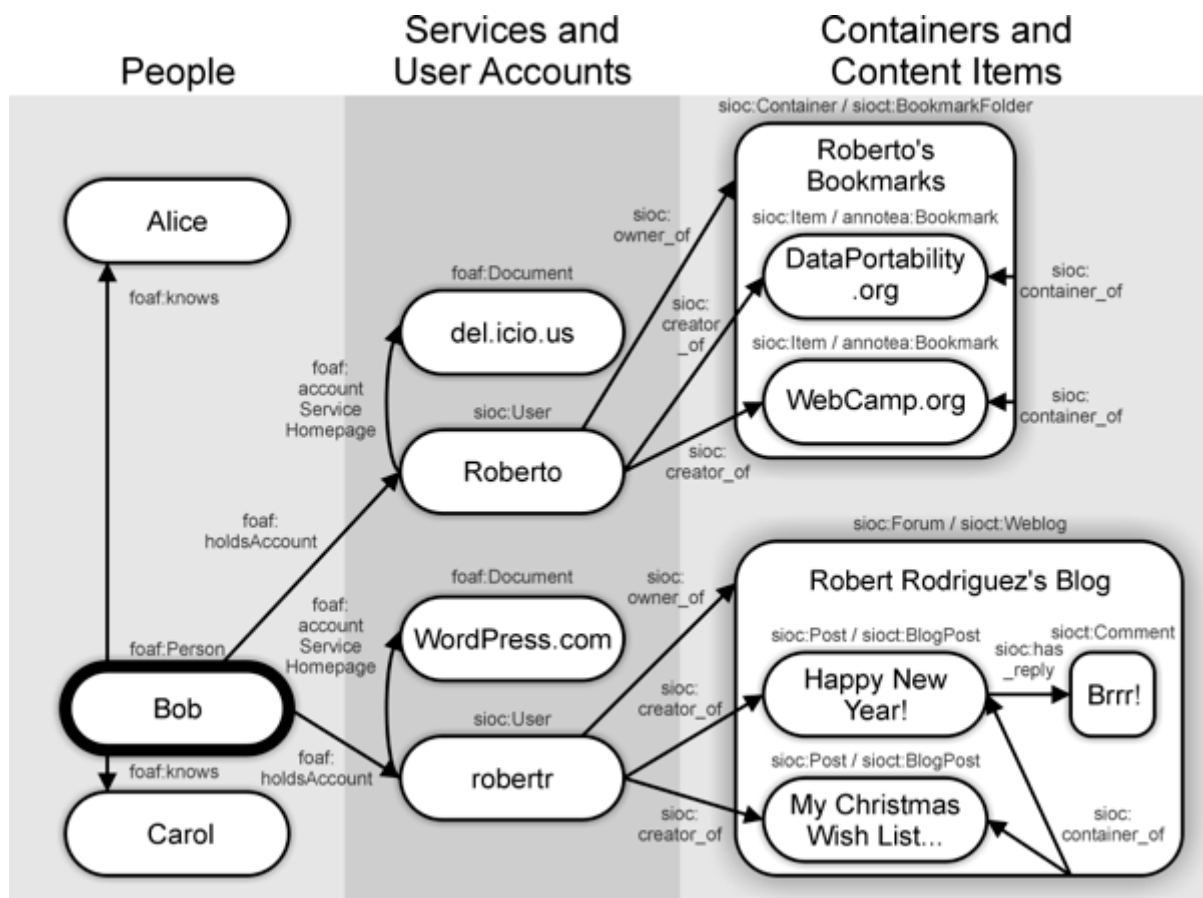


Figure 2-7 : Modeling social data with SIOC

Put differently, Figure 2-7 (W3C, 2008) shows that the FOAF Person is the owner of several sioc:User accounts. Foaf:Person can own many sioc:User accounts via the foaf:holdsOnlineAccount relationship. Figure 2-8 demonstrates this further showing the FOAF person holding online accounts on Usenet Groups, Weblog, Bulletin Board, Mailing list,

etc. It further shows the relation of SIOC with Simple Knowledge Organization System (SKOS).

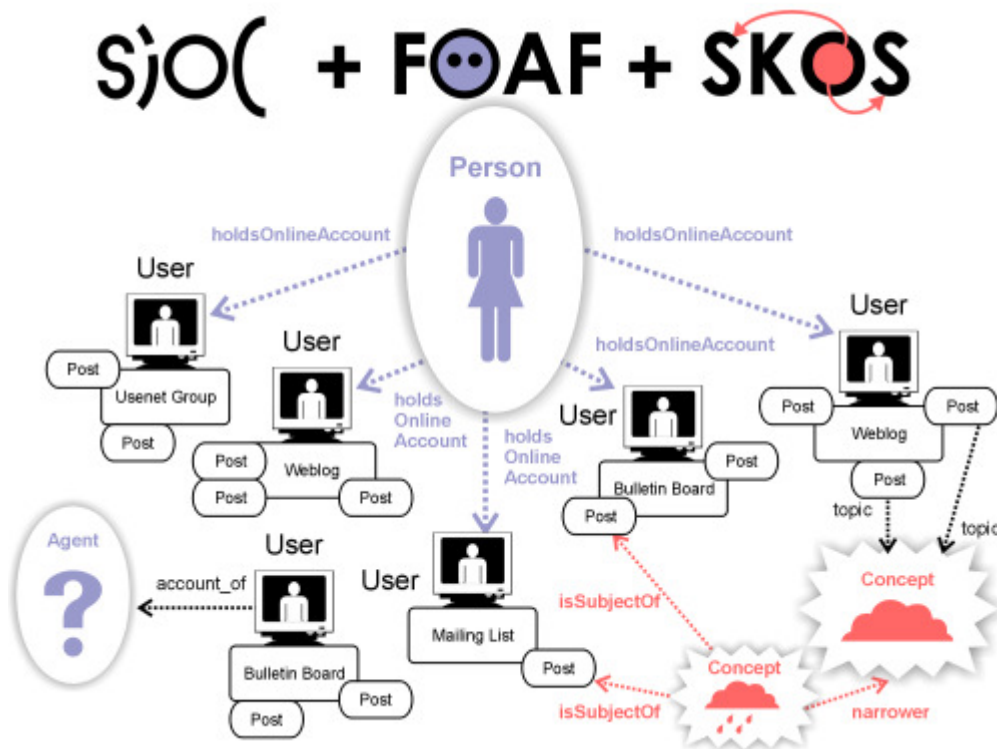


Figure 2-8 : SIOC FOAF Relation²⁵

2.3.3 FOAF and SIOC in use

Having been accepted as a W3C Member Submission, SIOC and FOAF are used in tens of applications from Web 2.0 to Life Sciences. Various SIOC exporters have been developed for existing applications, such as WordPress²⁶, Drupal and vBulletin. It has gone further to be used for mobile phone applications such as part of their data portability drive, the mobile phonebook service ZYB announced that their latest release of ZYB allows FOAF and SIOC exports from user's profile pages²⁷. In the following section we discuss Semantic Social Networks and MyExperiment, a project with a Social Networking component using FOAF and SIOC.

²⁵ <http://www.w3.org/Submission/2007/02/Comment>

²⁶ <http://sioc-project.org/wordpress>

²⁷ <http://pimpmy.zyb.com/post/2008/06/23/FOAF-and-SIOC-are-now-available.aspx>

2.3.3.1 MyExperiment

MyExperiment is a collaborative environment where scientists can safely publish their workflows and experiment plans, share them with groups and find those of others. Its data model has three main underlying features namely Content Management, Social Networking and Object Annotation. The myExperiment Ontology borrows terms from a number of well-known ontologies/schemas: (Goble, et al., 2007)

- Dublin Core to provide common metadata properties.
- FOAF and SIOC to describe the social network.
- Creative Commons to define licenses for shared objects.
- Open Archives Initiative Object Reuse and Exchange (OAI-ORE) to support the aggregation of resources.

myExperiment borrows terms in a number of ways:

- Equivalence relationships for properties or classes, (e.g. Group is equivalent to SIOC's UserGroup).
- Subclass/subproperty relationships, (e.g. email property is a subproperty of FOAF's mbox).
- Required properties in class restrictions, (e.g. an uploaded Contribution must have a Creative Commons license property).

The myExperiment ontology currently has 10 modules among which the “Base Module” that provides the base elements required by myExperiment for content management, social networking and object annotation. The base module depicted in Figure 2-9 uses the SIOC, FOAF, Creative Commons and Dublin Core Ontologies.

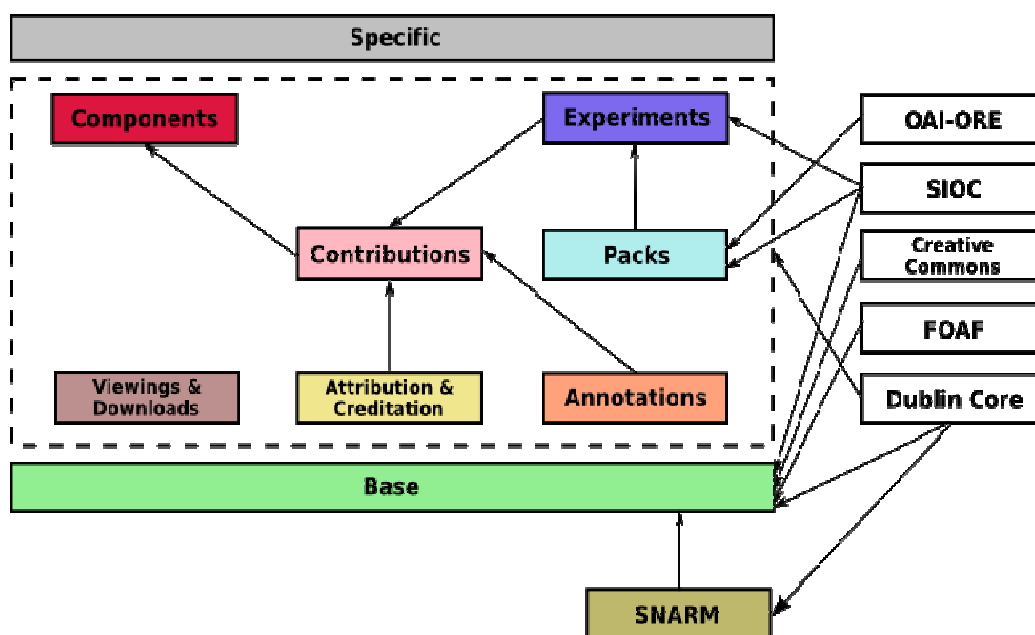


Figure 2-9 : MyExperiment Ontology Modules Architecture

2.3.3.2 Semantic Social Network

A semantic social network is the result of the application of Semantic Web technologies to social network technologies and online social media (Pollock, 2009). As thus the use of RDF, FOAF, SIOC and other ontologies will be evident in semantic social networks (Guillaume, et al., 2009). Downes states that “Two types of technologies are about to merge. The technologies are content syndication, used by blogging websites around the world, and social networking, employed by sites such as Friendster and Orkut. They will merge to create a new type of internet, a network within a network, and in so doing reshape the internet as we know it”²⁸.

2.4 Facebook Open Graph

Despite the initiatives like FOAF and SIOC discussed in the previous section, we are still seeing more “closed” online communities (Online communities where sharing of data is done within the confines of that online community). Though this is the state of affairs, Facebook’s launch of “Open Graph” on April 21, 2010 is a sign of hope that one day these communities may open up and adopt open standards like FOAF and SIOC. As Hendler and Berners-Lee state “A large-scale system may have emergent properties not predictable by

²⁸ <http://www.downes.ca/post/46>

analyzing micro technical and/or social effects (Hendler, et al., 2008).” It is still early to predict the emergent properties of Open Graph that is seemingly still a micro initiative but with big promise. By providing a ‘Like’ button that developers can add to any website, for any content or subject, Facebook is becoming the central hub for its user (Bryant, 2010). At the center of the Open Graph is the person who shares interests (likes) from different online services. Zuckerberg states that “This next version of Facebook Platform puts people at the center of the web. It lets you shape your experiences online and make them more social. For example, if you like a band on Pandora, that information can become part of the graph so that later if you visit a concert site, the site can tell you when the band you like is coming to your area. The power of the open graph is that it helps to create a smarter, personalized web that gets better with every action taken (Zuckerberg, 2010).”

When the idea of Open Graph API (Treadaway, et al., 2010) was introduced in 2009, Nick O’Neil stated that “Facebook will become the primary authentication hub for social experiences around the web. This authentication process makes Facebook the center of user identities and yesterday’s announcement of opening up developer access to emails means Facebook has literally become a wrapper for OpenID (O’Neill, 2009).” With many website administrators adding Facebook “Like” button on their websites and hence joining the Open Graph we may be headed to the macro property of seeing Facebook as the Authentication Hub. What we would then need to observe is if Facebook will adopt Semantic approaches such as FOAF to represent the person who is actually at the center of the Open Graph. Additionally, it will be informative to ascertain whether and if this adoption can result into something like “Facebook OpenID” as a defector login identifier borrowing from the OpenID project introduced earlier in this thesis. Further macro implication will arise if Facebook will adopt SIOC at some stage.

2.5 XML and X3D

The Extensible Markup Language (XML) introduced earlier is a plain-text format used by many Web languages including Hypertext Markup Language (HTML). XML is used to define other data-oriented Languages thus XML is not a language by itself, rather it is a language about languages, a metalanguage that is also suited for X3D.

X3D is an abbreviation for Extensible 3D, an ISO-approved standard formulated. The Web3D Consortium defines and develops the X3D royalty-free open standards file format and runtime architecture to represent and communicate 3D scenes (Web3D Consortium, 2010). The development of 3D web-based graphics has evolved from its beginnings with Virtual Reality Modeling Language (VRML) to X3D. In recent years, advances in real time graphics and networking technology, and most notably the emergence of the Extensible Markup Language XML, have heavily influenced the development of the ISO-approved, freely available, X3D open standard.

Rather than a language and more than a file format, X3D specifies runtime architecture to represent 3D scenes and objects. This runtime architecture is based on a scene graph, which models the entire 3D scene in one directed acyclic graph²⁹. X3D is a scene-graph architecture and file-format encoding that improves on the VRML international standard (formally listed as ISO/IEC 14772-1:1997 but frequently called VRML 2 or VRML97. X3D uses XML to express the geometry and behavior (Brutzman, et al., 2007). One of the greatest strengths of X3D is defining how a software browser ought to draw 3D images and interact with users, rather than trying to tell browser-building programmers exactly how to write their high-performance software.

²⁹ http://www.h3dapi.org/modules/mediawiki/index.php/Beginner_X3D_and_H3DAPI

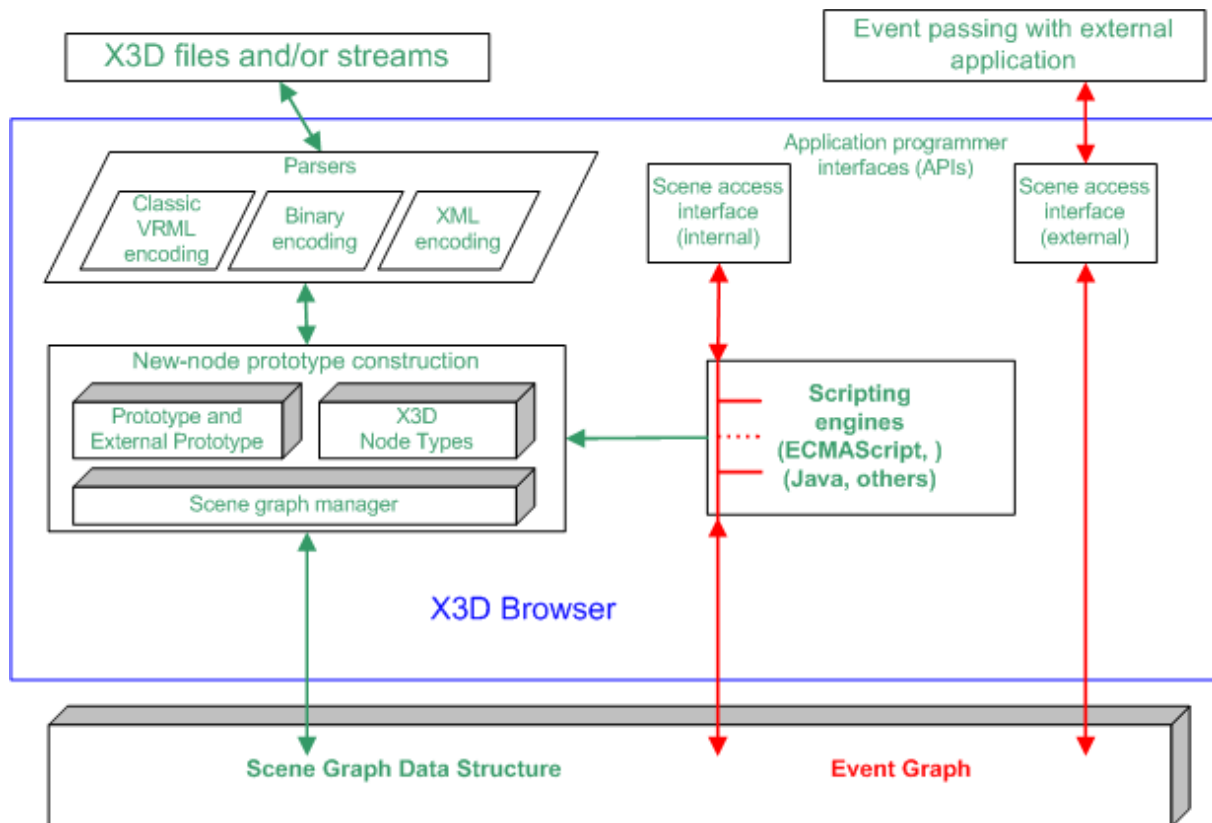


Figure 2-10 : X3D architecture

The example architecture of a sample X3D browser, illustrated in Figure 2-10 (Brutzman, et al., 2007), consists of file parsing and rendering and animation and dynamic scene graph manipulation. Browsers and applications that implement X3D capabilities can use any software architecture as long as the end results are visually and functionally correct (Daly, et al., 2007). X3D strikes an effective balance between lightweight, efficient Web-based delivery and sophisticated computation intensive applications by allowing authors to precisely define what capabilities are needed in each scene.

X3D browsers parse (read) X3D scene models and render (draw) them. They also provide simulation capabilities for animation and user interaction. X3D scene files have a common file structure consisting of

- File header (XML, ClassicVRML, and Compressed Binary)
- X3D header statement
- Profile statement - Profiles are predefined collections of components and components are predefined collections of nodes

- Component statements (optional)
- Meta statements (optional) - Meta statements provide information about the X3D scene
- X3D root node
- X3D scene graph child nodes

Figure 2-11 represents the popular X3D Hello World example in which the concepts mentioned above are observed. Between the head tags (<head>...</head>) there are meta statements to provide information such as creator, creation date and more about the X3D scene. Between the scene tags (<scene>...</scene>), the Group node is used to collect and select other nodes and the Transform node, as a grouping node, is used to define translation, rotation and scale. The viewpoint node defines the center of rotation from which users will explore the X3D scene. The shape node is being used as a container node for a single piece of geometry such as sphere or text. The appearance of the geometric object sphere and text is defined with the Appearance node in which visual surface properties like fill color properties and surface texture properties are detailed. Figure 2-12 shows how the X3D Hello World example looks like when viewed in a browser.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.2//EN" "http://www.web3d.org/specifications/x3d-3.2.dtd">
<X3D profile='Immersive' version='3.2' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.2.xsd'>
  <head>
    <meta content='HelloWorld.x3d' name='title'/>
    <meta content='Simple X3D example' name='description'/>
    <meta content='30 October 2000' name='created'/>
    <meta content='30 May 2009' name='modified'/>
    <meta content='Don Brutzman' name='creator'/>
    <meta content='http://www.web3D.org' name='reference'/>
    <meta content='http://x3dGraphics.com' name='reference'/>
    <meta content='http://www.web3d.org/x3d/content/examples/HelloWorld.x3d' name='identifier'/>
    <meta content='http://www.web3d.org/x3d/content/examples/HelloWorldTall.png' name='image'/>
    <meta content='http://www.web3d.org/x3d/content/examples/license.html' name='license'/>
    <meta content='X3D-Edit 3.2, https://savage.nps.edu/X3D-Edit' name='generator'/>
  </head>
  <Scene>
    <!-- Example scene to illustrate X3D nodes and fields (XML elements and attributes) -->
    <Group>
      <Viewpoint centerOfRotation='0 -1 0' description='Hello world!' position='0 -1 7'/>
      <Transform rotation='0 1 0 3'>
        <Shape>
          <Sphere/>
          <Appearance>
            <Material diffuseColor='0 0.5 1'/>
          </Appearance>
        </Shape>
      </Transform>
    </Group>
  </Scene>
</X3D>
```

```

    <ImageTexture url=""earth-topo.png" "earth-topo.jpg" "earth-topo-small.gif"
"http://www.web3d.org/x3d/content/examples/Basic/earth-topo.png"
"http://www.web3d.org/x3d/content/examples/Basic/earth-topo.jpg"
"http://www.web3d.org/x3d/content/examples/Basic/earth-topo-small.gif"/>
  </Appearance>
</Shape>
</Transform>
<Transform translation='0 -2 0'>
  <Shape>
    <Text solid='false' string=""Hello" "world!"">
      <FontStyle justify=""MIDDLE" "MIDDLE""/>
    </Text>
    <Appearance>
      <Material diffuseColor='0.1 0.5 1'/>
    </Appearance>
  </Shape>
</Transform>
</Group>
</Scene>
</X3D>

```

Figure 2-11 : X3D Hello World ³⁰

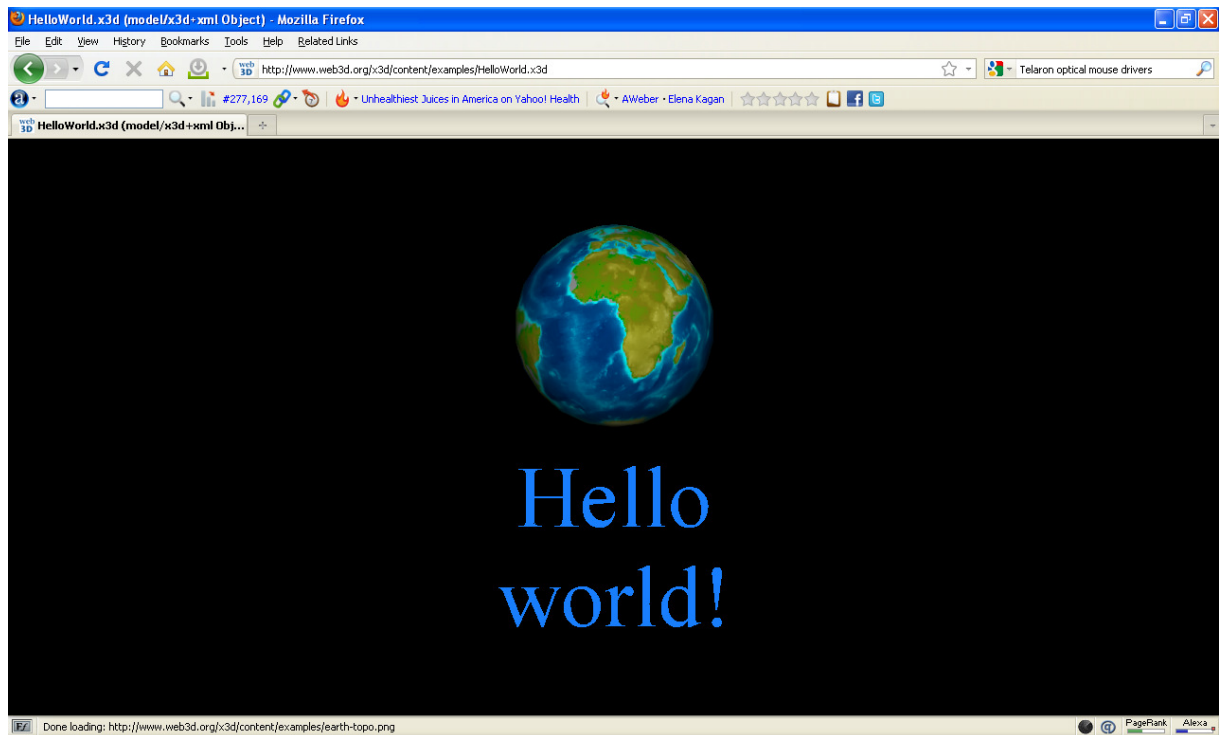


Figure 2-12 : X3D Hello World in Browser

³⁰ <http://www.web3d.org/x3d/content/examples/HelloWorld.x3d>

2.6 XSL Transformations

We have seen that X3D is an XML based language and the FOAF and SIOC data that we shall work with will be represented in RDF and XML. We shall need to convert data from an XML file to be represented in X3D. XSLT comes handy to enable transformation of XML documents.

XSLT (XSL Transformations) is a declarative, XML-based language used for the transformation of XML documents into XML, HTML, WML, TXT and other document formats. The original document is not changed, rather a new document is created based on the content of an existing one (W3C, 1999).

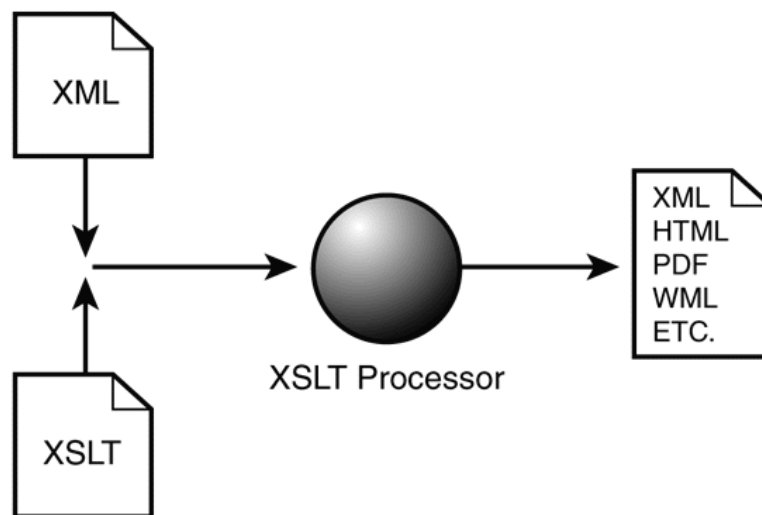


Figure 2-13 : XSLT Transformation³¹

The XSLT processing model involves:

- One or more XML *source* documents.
- One or more XSLT *stylesheet* modules.
- The XSLT template processing engine (the *processor*)
- One or more *result* documents.

As shown in Figure 2-13 , the XSLT processor ordinarily takes two input documents an XML source document and an XSLT stylesheet and produces an output document. The XSLT

³¹ http://www.xmlfiles.com/articles/sample_chapters/sams_xmlforaspnet/default.asp

stylesheet contains a collection of template rules: instructions and other directives that guide the processor in the production of the output document. The W3C provides the following statement about templates “A stylesheet contains a set of template rules. A template rule has two parts: a pattern which is matched against nodes in the source tree and a template which can be instantiated to form part of the result tree. This allows a stylesheet to be applicable to a wide class of documents that have similar source tree structures” (W3C, 1999).

To illustrate this better the example Figure 2-14 from (Muench, 2000)

Original XML	Applied XSLT	Result
<pre><!-- Emp.xml --> <ROWSET> <ROW num="1"> <EMPNO>7839</EMPNO> <ENAME>KING</ENAME> </ROW> <ROW num="2"> <EMPNO>7788</EMPNO> <ENAME>SCOTT</ENAME> </ROW> </ROWSET></pre>	<pre><xsl:template match="ROWSET/ROW[ENAME]"> <Employee id="NX-{EMPNO}"> <xsl:value-of select="ENAME"/> </Employee> </xsl:template></pre>	<pre><Employee id="NX-7839" > KING </Employee> <Employee id="NX-7788" > SCOTT </Employee></pre>

Figure 2-14 : XSLT Example

In the example in Figure 2-14 the following three things occur:

1. Literal result elements and attributes in the template are created in the result tree. Result elements and attributes that are not from the XSLT namespace are considered “literal” since they are constructed as is in the result tree. In the example the <Employee> element and its id attribute are created.
2. Any attribute value templates of the form {XPathExpr} contained within literal attribute values are replaced by the value of their XPath expression. In the example, the {EMPNO} inside the literal attribute value NX-{EMPNO} is replaced by the value of the EMPNO XPath expression. This evaluates to 7839, so the final value for the id attribute in the result tree is NX-7839.
3. Any elements in the XSLT namespace are processed in document order. The <xsl:value-of> element is processed and is replaced by a text node containing the string value of the XPath

expression in its select attribute—in this case, KING. The basic operation can be summarized as follows: when a node in the source matches a rule’s pattern, the content of that rule is created in the result tree.

2.7 Virtual World

The Web 2.0 communities discussed earlier limit user interaction to a 2D environment. It is not possible to simulate a clear test of someone in terms of things such as color preference, preferred dress style, a live dance party etc. Virtual Worlds, that we aim to generate for online communities fill the gap and enhance user interaction by simulating objects to be as close as possible to real world objects.

A Virtual World is a genre of online community that often takes the form of a computer-based simulated environment, through which users can interact with one another and use and create objects (Bishop, 2009). Virtual Worlds are intended for its users to inhabit and interact, and the term today has become synonymous with interactive 3D virtual environments. The users take the form of avatars visible to others graphically (Cook, 2009).

Virtual Worlds are currently being deployed in web social applications such as “Second Life”³², in education for virtual learning environments. In the following section we explore some of the existing Virtual World services and tools.

2.7.1 Second Life

“Second Life” (SL) is a free 3D Virtual World imagined and created by its users called Residents. Residents can explore, meet other residents, socialize, participate in individual and group activities, and create and trade virtual property and services with one another, or travel throughout the world (which residents refer to as "the grid") (Rymaszewski, et al., 2006). Figure 2-15 shows an insight into the Second Life interface using a free Open Source client program called the Viewer that enables the Residents to control their in-world avatars, interact with each other via Instant Message, create content, buy and sell objects, access multimedia content and to navigate around the virtual environment ³³.

³² <http://secondlife.com>

³³ http://lindenlab.com/pressroom/releases/01_08_07

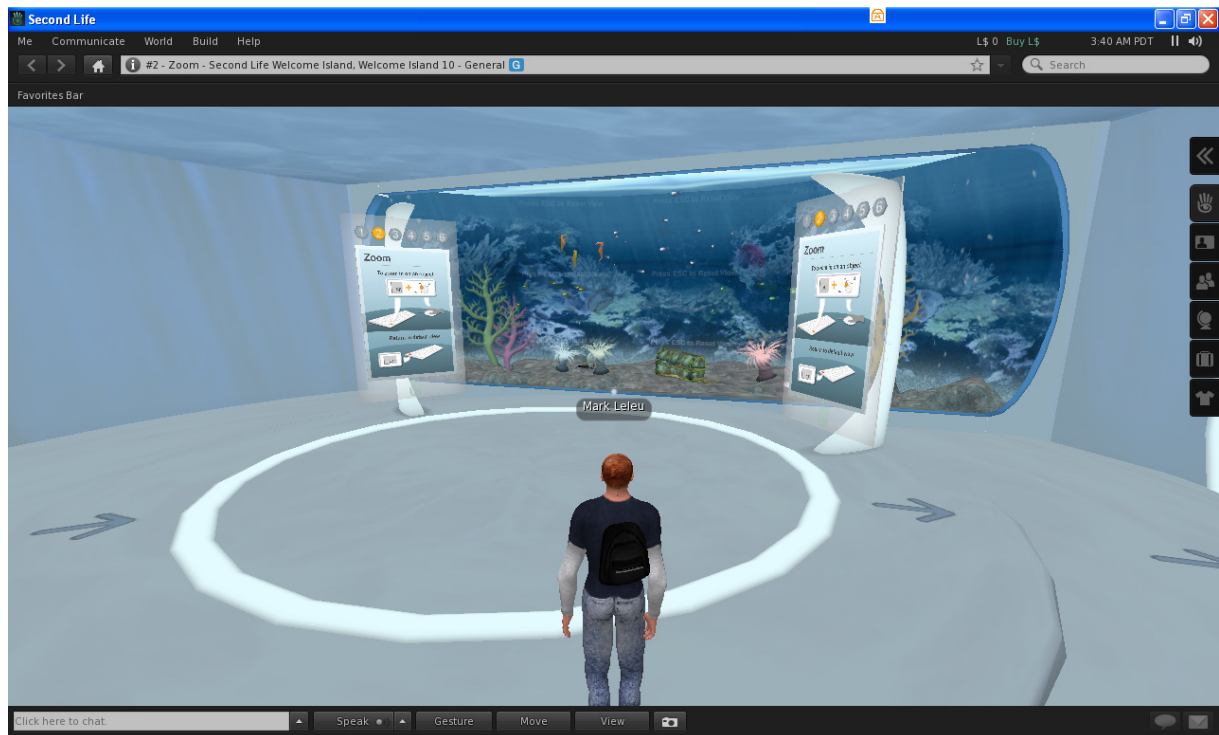


Figure 2-15 : Second Life Insight

Although second life offers a Virtual World experience, it is limited to interacting with predefined communities. Therefore, it is impossible to view and interact with your friends from other online communities such as Facebook. In addition, its viewer is not directly intergraded in the browser.

2.7.2 Vivaty

Vivaty³⁴ is a web-based Virtual World platform that is built entirely on open standards such as X3D. Vivaty content is broken into three categories namely: scenes, objects and avatars. Scenes are 3D environments that Vivaty users can explore, socialize in and customize. Scenes can be realistic or fantasy-based enabling avatars to hang out in style and move around comfortably. Objects are things that can be added to Vivaty scenes. Objects can be as simple as a brick, or as complicated as a YouTube video player or Facebook friends list. Vivaty content is deployed over the web, and rendered in real time thus it cannot afford to use extremely high resolution meshes or textures. A compromise on high quality, low resolution creation, is recommended (Vivaty, 2007).

³⁴ <http://developer.vivaty.com/>

Vivaty offers Vivaty Studio, a free tool that is optimized for creating content. Vivaty recommends that developers who use other development tools such as 3D Studio Max or Maya to use COLLADA an interchange format to import scenes to Vivaty studio. Vivaty enables the export of X3D format files. Figure 2-16 shows four views (Top, Front, Right and Iso View) of a Box in Vivaty Studio which when viewed in a browser or Vivaty player appears as in Figure 2-17. The source file of this 3D box is shown in an extract in Figure 2-18 in which we can see that the shape node is being used as a container node for box which is a single piece of geometry.

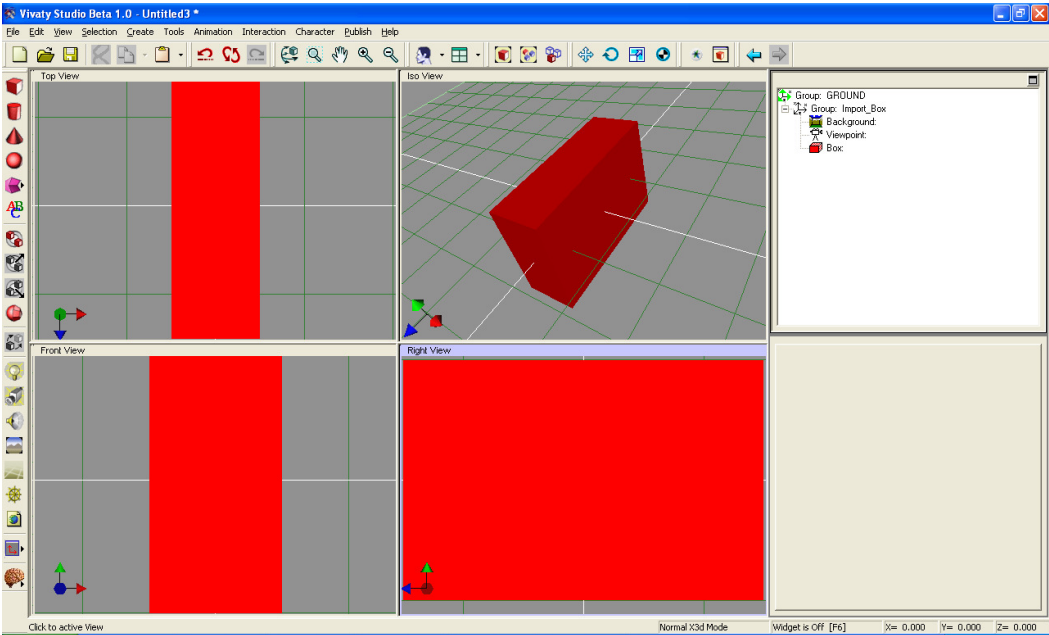


Figure 2-16 : Vivaty Studio

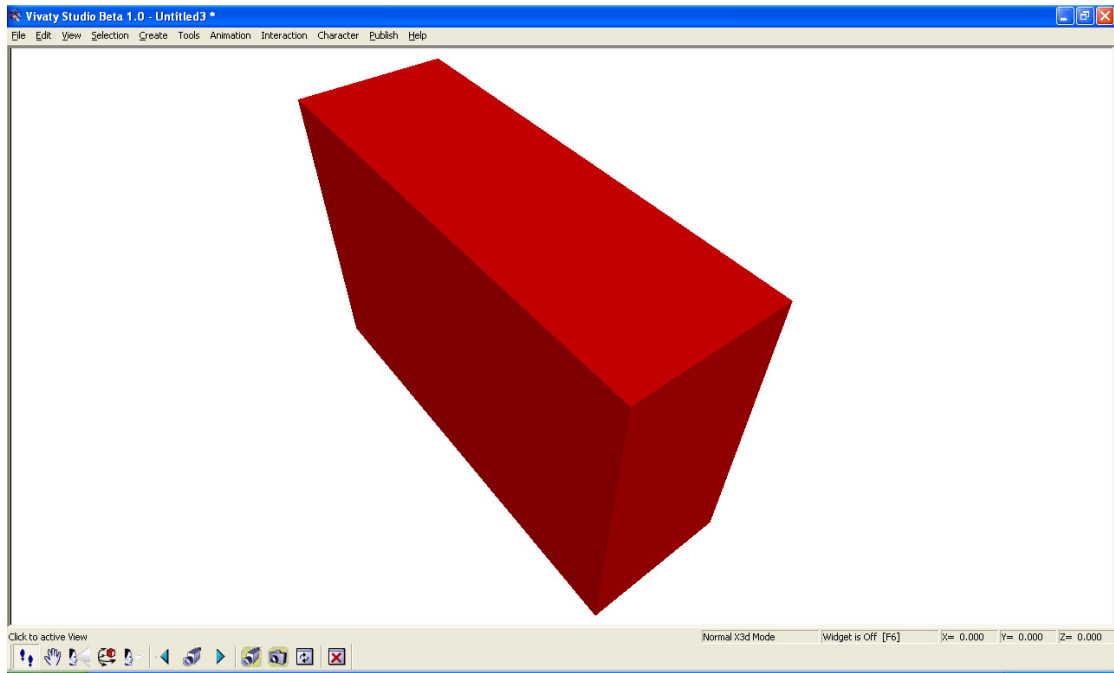


Figure 2-17 : Vivaty Player

```

<Scene>
<WorldInfo
  title='Untitled3'
  info='"This Web3D Content was created with Vivaty Studio, a Web3D
authoring tool"
  "www.mediamachines.com"' />
<Background
  skyAngle=''
  skyColor='
  1 1 1' />
<Viewpoint
  description='Book View'
  jump='true'
  fieldOfView='0.785'
  position='-1.81 3.12 2.59'
  orientation='-.747 -.624 -.231 1.05' />
<Shape>
  <Appearance
    containerField='appearance'>
    <Material DEF='Red'
      containerField='material'
      ambientIntensity='0.200'
      shininess='0.200'
      diffuseColor='1 0 0' />
    </Appearance>
    <Box
      containerField='geometry'
      size='1 2 3' />
  </Shape>
</Scene>
</X3D>

```

Figure 2-18 : Box X3D

Vivaty Studio enables us to experiment with 3D scenes and possible capabilities. It is more of an authoring environment and not a tool or Virtual World to do on the fly transformations of 2D content to 3D.

2.7.3 ExitReality

ExitReality, launched on 18th September 2008 is a 3D Virtual World service that transforms normal 2D web pages to 3D Virtual Worlds. With a big dream to transform the entire web into 3D, “ExitReality goes far and beyond any other 3D destination ever created. With Second Life and World of Warcraft, users visit the one destination for a virtual experience, but ExitReality allows you to view not just that one website, but the entire World Wide Web in 3D,” said founder of ExitReality Danny Stefanic (Smart PR Corporation, 2008).

ExitReality requires an add-on program for your browser (Firefox or Explorer) to incorporate 3D functionality. Quoting Danny Stefanic, “The free Internet plug-in takes only 20 seconds to download, but opens up the user to a whole new universe. It transports you to a world where social network pages such as Facebook become 3D apartments users are able to decorate; a world where YouTube is transformed into the world’s largest 3D cinema where people can watch movies with their friends; a world where photo albums become virtual galleries and music is played out of a jukebox. And it’s not a solo experience — the user can see and share experiences with their friends while chatting with them and other people at either their own website or the other billion web pages (Smart PR Corporation, 2008).”

With real hands on experience using ExitReality, we observe that ExitReality allows you to meet and socialize with other people who are visiting the same page you are visiting at that very moment. Though it has a big ambition to turn the entire web into 3D, ExitReality works best with Web 2.0 sites, such as MySpace, Facebook, YouTube and media sites like warnerbros.com. As seen in Figure 2-19, in line with the vision of the founder Danny Stefanic quoted earlier, we observe that it maps out MySpace or Facebook page into a 3D room where videos on the MySpace page will always appear on the television set that is in the room and any audios will be available on the jukebox and images plus Friends photos are in painting frames. In Figure 2-20, we can see that images and Videos are mapped on wall images in the warnerbros.com 3D Environment.



Figure 2-19 : ExitReality Facebook Space

Frames to be used to represent friends photos in the Facebook Room



Figure 2-20 : ExitReality WarnerBros.com

ExitReality offers the option to use different themes to represent a given 3D Environment. As illustrated in Figure 2-21 , users can select themes.

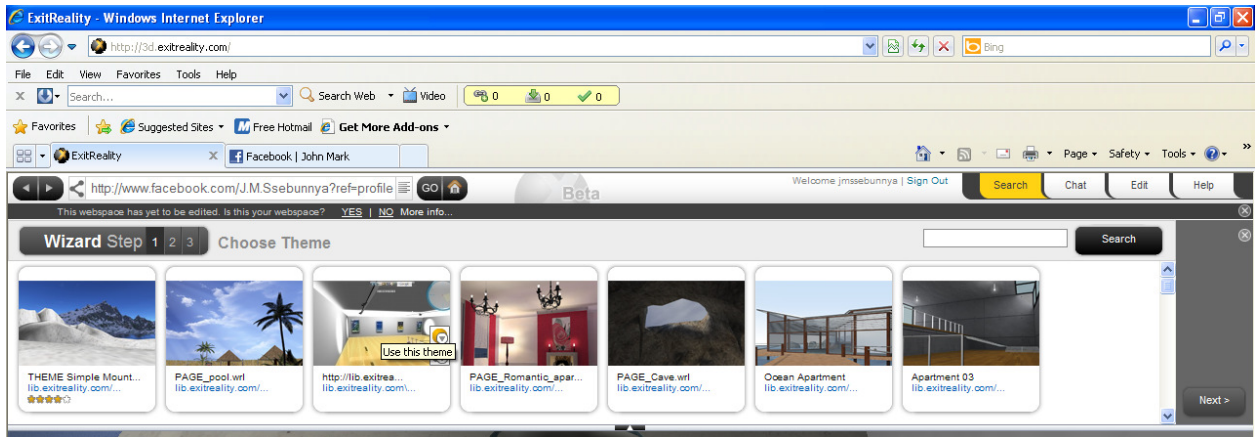


Figure 2-21 : ExitReality Selecting Themes

ExitReality has several advantages such as not requiring special graphics cards, needs a small download and has an interface that integrates with normal web browsers enabling viewing of regular 2D content in 3D space. However, it is not yet effectively achieving that. It will require all 2D websites to conform to certain designs or to have annotations that would help in successful mapping. A visit to cnncn.com in ExitReality reveals a very poor representation of this website in 3D as seen in Figure 2-22.

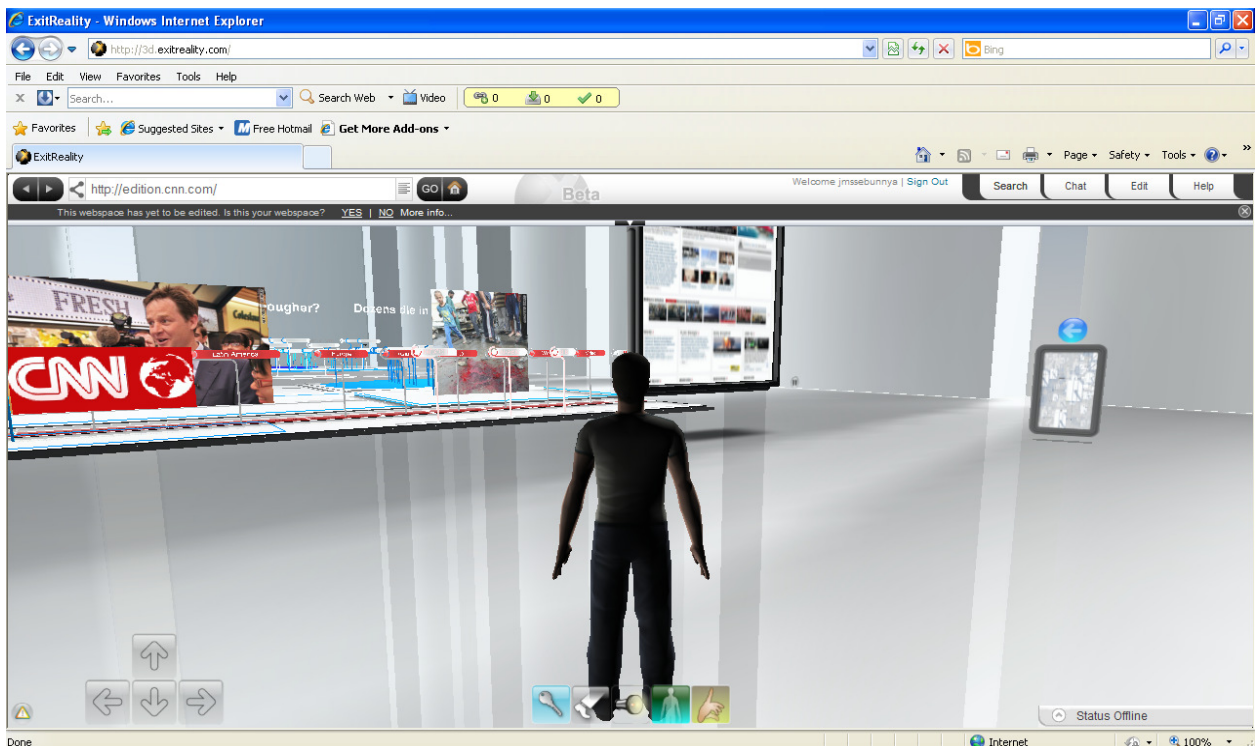


Figure 2-22 : ExitReality CNN.com

2.8 Website Transformation Approaches

Due to the need to either present website information on different devices or to a different audience with particular usability requirements such as the visually impaired, there is a need to sometimes transform a website from one format to another. This can be done with the transformation of content from one format into another i.e. Transcoding. Transcoding approaches are grouped into two namely:

1. Rule Based Transcoding: This is where an application of patterns or known statistics to web pages is done in order to extract the implicit information. Rule based transcoding systems do have the advantage that they can work with any website thus one set of rules fits all websites and so you can attempt to transcode anything in the web. As seen in section 2.7.3 ExitReality can transform any web page to a 3D Virtual World and we could see this approach as adhering to rule based transcoding. Though this is possible with ExitReality, it does a poor job of transcoding for the majority of web pages.

2. Semantic Transcoding: This is where systems that use semantic transcoding attempt to capture the meaning of the elements of a webpage by annotating all the elements that occur within the page. For example, if there is a menu contained within the webpage, then a comment is added to the source code that states "this is a menu". The transcoding system then has to look for the annotations and manipulate the content as it deems appropriate. The benefit of such systems is that they are incredibly accurate. A menu is clearly labeled as a menu, so therefore it cannot be anything else (unless the annotator made a mistake). However, the big disadvantage of this is that it is time consuming, tedious and sometimes impossible to annotate every webpage on every website in order for it to be transcoded³⁵.

In transforming 2D online communities to 3D Virtual Worlds, we need to understand the concept of transcoding. In the following section we present the Dante and SADie frameworks that are used for transcoding websites to allow accessibility for visually impaired users.

2.8.1 Dante

In the real world, visually impaired users travel looking ahead for obstacles by tapping a stick to get auditory feedback from the environment. The act of using a stick to probe the environment is similar to probing hyperlinks to see what the content at the link destination

³⁵ <http://hcv.cs.manchester.ac.uk/research/sadie/>

is. A stick could give feedback to obstacles or a clear path in the environment. Following a hyperlink either takes the user to irrelevant information or relevant information (a clear path) (Lunn Darren, 2006). On the web, most Web pages are designed for visual interaction so the mobility or ease of travel, of visually impaired Web travelers is reduced. Objects that support travel and mobility are not in an appropriate form for non visual interaction. Supported with an automated tool “Dante” aims to enhance the mobility of visually impaired Web travelers by annotating pages with a travel ontology that aims to encapsulate rich structural and navigational knowledge (Yeliz, et al., 2004).

As shown in the Dante architecture in Figure 2-23 , when a web page Document Object Model (DOM) is passed on, it is analyzed to identify travel objects in stage (1), the roles of the travel objects are discovered in stage (2), annotation of the identified objects by using a travel ontology is done in stage (3) and transformation of the page with respect to these annotations is done in stage (4) and then a new page is returned to the User Agent.

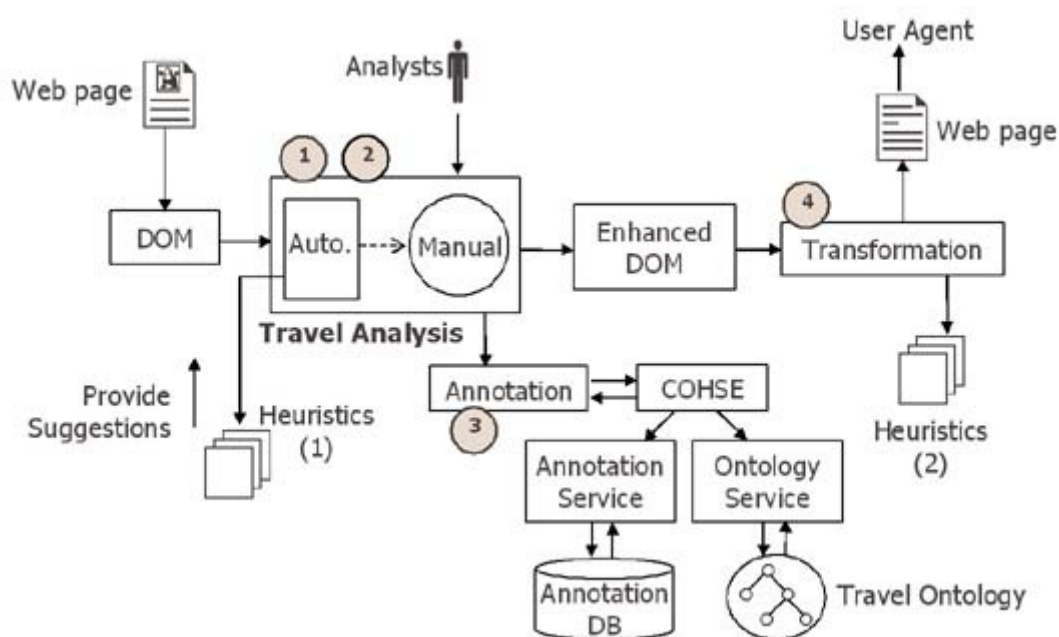


Figure 2-23: Dante³⁶

2.8.2 SADie

SADie³⁷ (Structural-Semantics for Accessibility and Device Independence) is a method of semantic transcoding of web pages in order to aid visually impaired users. SADie taps into

³⁶ <http://dante.cs.manchester.ac.uk/dante/danteArchitecture.jpg>

present day presentation techniques where a single CSS file describes the rendering of an entire website which can be many pages in a big website such as blogger.com. Sadie describes annotations on a single CSS file and hence can potentially transcode thousands of web pages on the website.

Shown in Figure 2-24 for a particular cascading style sheet (CSS), an extension to an upper-level ontology which represents the interface between web documents and the programmatic transformation mechanism is provided. This extension gives the particular characteristics of the elements appearing in that specific CSS. This extension is considered to be an annotation of the CSS elements implicitly encoded into the web document. This means that one ontology can be used to accurately transform every web document that references the CSS used to generate that ontology. Simply one ontology accurately transforms an entire site using generalized programmatic machinery able to cope with all sites using CSS (Harper, et al., 2007). In Figure 2-24, an XHTML document at stage (2) is described with a CSS at stage (1). The CSS at stage (1) is used to create concepts in the ontology at stage (3). The SADie application at stage (4) then retrieves the ontology much like Firefox retrieves the CSS document. The ontology is passed to the ontology service and SADie can then ask questions based on the actions required at stage (5).

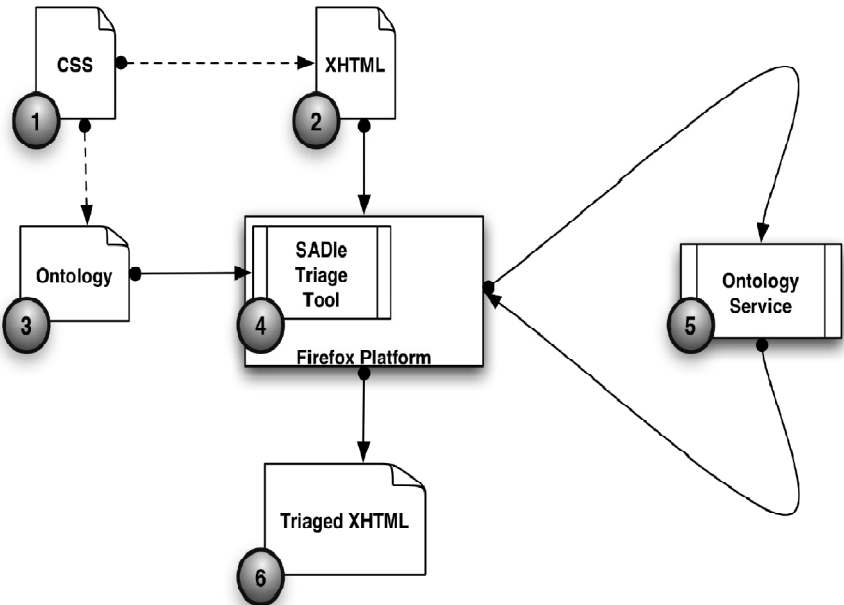


Figure 2-24 : SADie

³⁷ <http://hcw.cs.manchester.ac.uk/research/sadie/>

As discussed earlier, SADle transcoding is based on CSS and works well for particular web pages that have CSS defined. SADle's approach is not about gathering data from different communities or websites and presenting this data in a complete new format. It is only concerned with the presentation aspect and as thus its core idea revolves around defining CSS concepts with ontology.

3 FOAF and SIOC Based 3D Approach

In Chapter 2 we discussed current Virtual Worlds such as ExitReality and Second Life. We observed that ExitReality transforms any 2D web page to 3D using some approach similar to rule based transcoding. Second Life is focused on creating a predesigned world that residents interact with. We also discussed initiatives like Sadie and Dante which aim at adaptation of content from one format into another, using semantic approaches that deploy annotations. All the initiatives discussed are focused on getting individual pages or individual communities and representing them in another format. Our aim is to be able to capture content from different online communities and represent this data in a 3D Virtual World. To achieve this, we propose the “FOAF and SIOC Based 3D Approach” shown in Figure 3-1 whose major phases are:

1. Social Media Mining: This phase brings together FOAF Mining, SIOC Mining and Integration stages to extract FOAF data from a provided FOAF file and from other communities, to extract SIOC data and to integrate the FOAF and SIOC data into one file.
 - a. FOAF Mining: This stage starts with a FOAF profile from which relevant data is extracted and stored in a new XML file which we shall call “FOAF Data”. The extracted data will be used as the basis for the person profile. The communities that allow FOAF exportation will be visited (mined) to get an extensive list of friends. In collecting this list of friends there could be a repetition of same friends that are members of different communities. This can be ironed out manually.
 - b. SIOC Mining: At this stage, data will be imported from communities that support exporting RDF data into a format based on the SIOC ontology. This data will represent contributions (Social Media Contributions) to the online communities and will be stored in a “SIOC Data” XML file.
 - c. Integration: Having captured FOAF and SIOC data, it will be integrated into one XML file which we shall call “Person Communities”. This file is an instantiation based on an Ontology which we call the Person Communities Ontology.

FOAF and SIOC Based 3D Approach

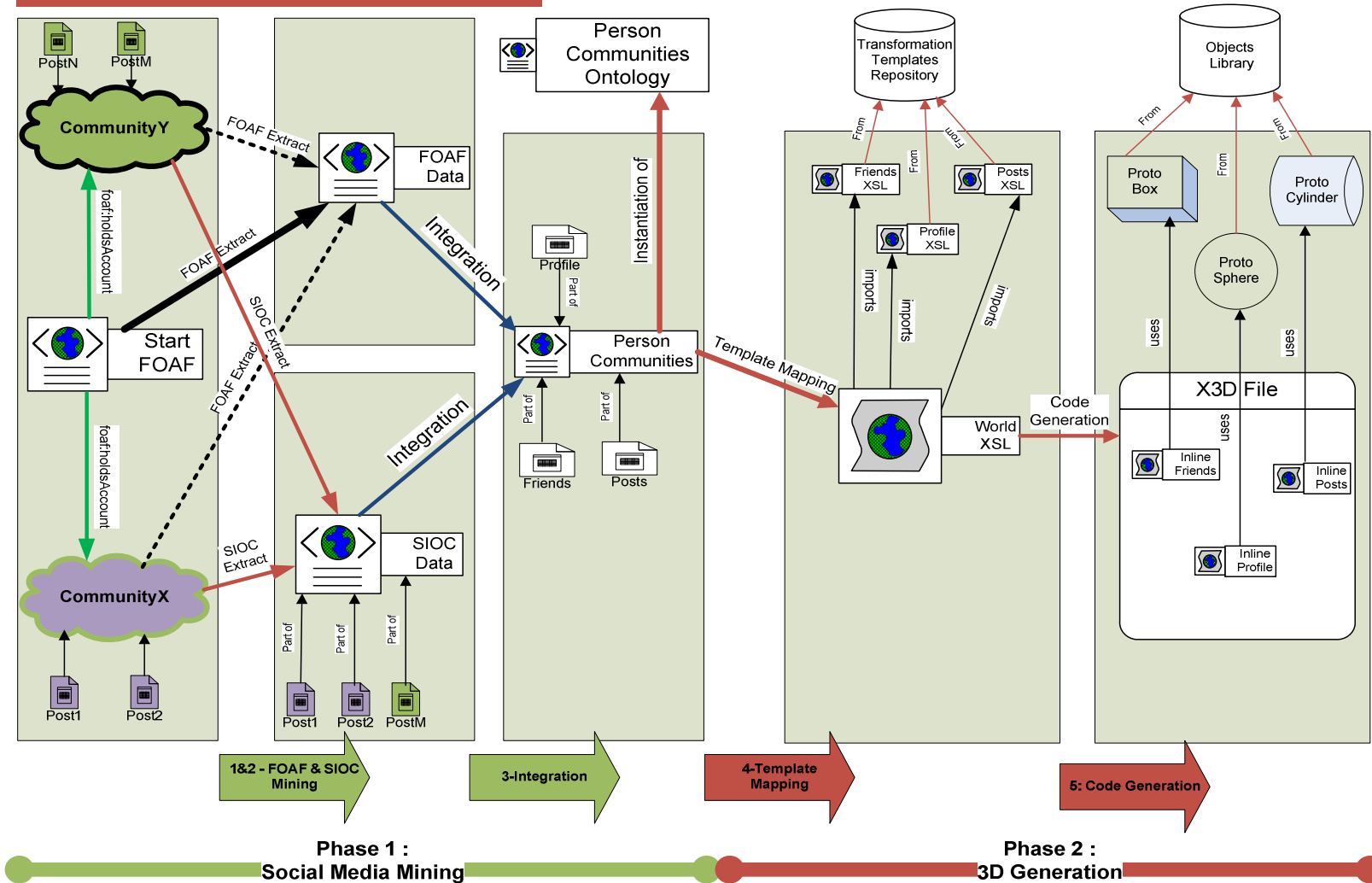


Figure 3-1 : Transformation Stages

2. 3D Transformation: This phase is divided into Template Mapping and Code Generation stages. A choice of templates to represent FOAF and SIOC data concepts is done and X3D code is generated.
 - a. Template Mapping: Templates will be collected from a repository of templates. The user will choose which template to use to represent different concepts in the 3D world.
 - b. Code Generation: Using XSLT, transformation will be performed to convert the “Person Communities” XML to the X3D representation.

The approach deploys concepts such as FOAF and SIOC ontologies, XSLT and X3D that we have discussed in Chapter 2. The transformation will be easy to change based on new templates that can be applied. The approach is web based so we do not have to use an external tool in which to import files for transformation. Put diffidently, the whole operation occurs in the browser and on the web server. In the following sections we discuss the approach in more detail.

3.1 Social Media Mining

As stated earlier, a person can hold accounts on several online communities to which he/she actively participates. The person will have different contributions and different networks of friends on these communities. The contributions and the networks of friends are the social media to which the person broadcasts. In order to capture data about the person, we shall need to mine the communities and retrieve the friends list and contributions. Berners-Lee put it, “I express my network in a FOAF file, and that is a start of the revolution (Pollock, 2009).” In this Social Media Mining Phase we shall start with the FOAF profile the person will provide and follow through to the other communities the person is a member of. In doing so we shall perform FOAF Mining, SIOC Mining and finally integration of the extracted data into the Person Communities Ontology.

3.1.1 FOAF Mining

In this step the user will provide a FOAF Profile accessible from a web link. The FOAF profile can either be manually created or a using an online tool such as FOAF-a-Matic we introduced in Chapter 2. Figure 3-1 shows a “Start FOAF” file from which we can see a link to

CommunityX and CommunityY via the foaf:holdsAccount relationship. In the “Start FOAF” extract shown in Figure 3-2, the basic attributes such as given name, family name, etc. are represented. A list of accounts (CommunityX, CommunityY, Facebook, Twitter) the person holds is available and a list of friends (Friend_1, Friend_A, Friend_M) is also added.

```

<foaf:Person>
<foaf:name>PersonFirstName PersonLastName</foaf:name>
<foaf:title>Mr</foaf:title>
<foaf:givenname>PersonFirstName</foaf:givenname>
<foaf:family_name>PersonLastName</foaf:family_name>
<foaf:nick>PersonNickName</foaf:nick>
<foaf:mbox_sha1sum>f2f9c01d3b208297c02d25c301bdaafc2472cbe4</foaf:mbox_sha1sum>

<foaf:holdsAccount>
  <foaf:OnlineAccount>
    <foaf:accountServiceHomepage rdf:resource="http://www.facebook.com/"/>
    <foaf:accountProfilePage rdf:resource="http://www.facebook.com/Person1"/>
    <foaf:accountName>J.M.Ssebunnya</foaf:accountName>
  </foaf:OnlineAccount>
    <foaf:OnlineAccount>
      <foaf:accountServiceHomepage rdf:resource="http://communityX.info/"/>
      <foaf:accountProfilePage rdf:resource="http://communityX.com/person1"/>
      <foaf:accountName>Admin</foaf:accountName>
    </foaf:OnlineAccount>
      <foaf:OnlineAccount>
        <foaf:accountServiceHomepage rdf:resource="http://communityY.com/"/>
        <foaf:accountProfilePage rdf:resource="http://communityY.com/person1"/>
        <foaf:accountName>nico</foaf:accountName>
      </foaf:OnlineAccount>
        <foaf:OnlineAccount>
          <foaf:accountServiceHomepage rdf:resource="http://twitter.com/"/>
          <foaf:accountProfilePage rdf:resource="http://twitter.com/Person1"/>
          <foaf:accountName>jmssebunnya</foaf:accountName>
        </foaf:OnlineAccount>
      </foaf:holdsAccount>

<foaf:knows>
<foaf:Person>
<foaf:name>Friend_A</foaf:name>
<foaf:mbox_sha1sum>38960ae59d15b930e88494150888f0b393707b07</foaf:mbox_sha1sum></foaf:Person>
</foaf:knows>
<foaf:knows>
<foaf:Person>
<foaf:name>Friend_1</foaf:name>
<foaf:mbox_sha1sum>165fcb142f749273cdc904a0e97fa2c264bde6ee</foaf:mbox_sha1sum></foaf:Person></foaf:knows>
<foaf:knows>
<foaf:Person>
<foaf:name>Friend_M</foaf:name>
<foaf:mbox_sha1sum>8f514d1259a79d3aee9c0108162e481edf2bea7f</foaf:mbox_sha1sum></foaf:Person></foaf:knows>
</foaf:Person>

```

Figure 3-2 : Start FOAF Extract

When the user provides a FOAF profile link with this data in Figure 3-2, the relevant data will be extracted from the XML file (See Figure 3-1 “FOAF Extract” full arrow) and presented to the user, to be visualized. The extracted data will include concepts to represent the person attributes, the online accounts the person holds and a list of friends. These concepts would appear in the “FOAF Data” XML file in a structure such as Figure 3-3. This will be done using scripting or programming tools that deploy Document Object Model (DOM) functions, i.e., it can be performed using JAVA, ASP or PHP. Alternatively the transformation can be done using XSLT and server-end technology would be used to save the generated file.

```
<foaf_sioc_combined version="foaf_basic">
  <Person>
    <name></name>
    <givenname></givenname>
    <family_name></family_name>
    <birthday></birthday>
    <locality></locality>
    <region></region>
    <country></country>
    <gender></gender>
    <holdsAccount>
      <OnlineAccount>
        <accountServiceHomepage>
</accountServiceHomepage>
        <accountProfilePage></accountProfilePage>
        <accountName></accountName>
      </OnlineAccount>
    </holdsAccount>
    <knows>
      <Person>
        <name></name>
      </Person>
    </knows>
  </Person>
</foaf_sioc_combined>
```

Figure 3-3 : FOAF Data Structure

As much as the “Start FOAF” file provides data we need, it so happens that we may not have a comprehensive friends list since the network of friends could be spread in different communities. Figure 3-5 represents the foaf:holdAccount relations that are in the FOAF extract in Figure 3-2. In this figure we can see the friends from other communities who were not listed in the provided FOAF. As thus we may consider visiting some of the communities where the person holds an account (See Figure 3-1 “FOAF Extract” dotted arrow). If these communities allow exportation of FOAF data we extract this data, pick a friends list from it and add this list to our existing list of friends otherwise, the data in the Start FOAF file will be the data to work with. It is important to note that we are going to be faced with the issue of

similar friends appearing in different communities as seen in the Figure 3-5 with the “SameAs” link. This could be sorted out manually at this stage and result in the list as shown in Figure 3-4 though with better representation of this data an automated alternative may be worked out.

<p>Friends List from “Start FOAF” File Provided</p> <ol style="list-style-type: none"> 1. Friend_1 2. Friend_A 3. Friend_M 	<p>New friends list after other communities are visited and mined.</p> <ol style="list-style-type: none"> 1. Friend_1 2. Friend_2 3. Friend_A 4. Friend_B 5. Friend_C 6. Friend_M
---	---

Figure 3-4 : Friends List

At this stage a comprehensive list of friends will be added to the “FOAF Data” file. Hence the FOAF mining will end with the FOAF Data file having the profile, friends and online accounts information.

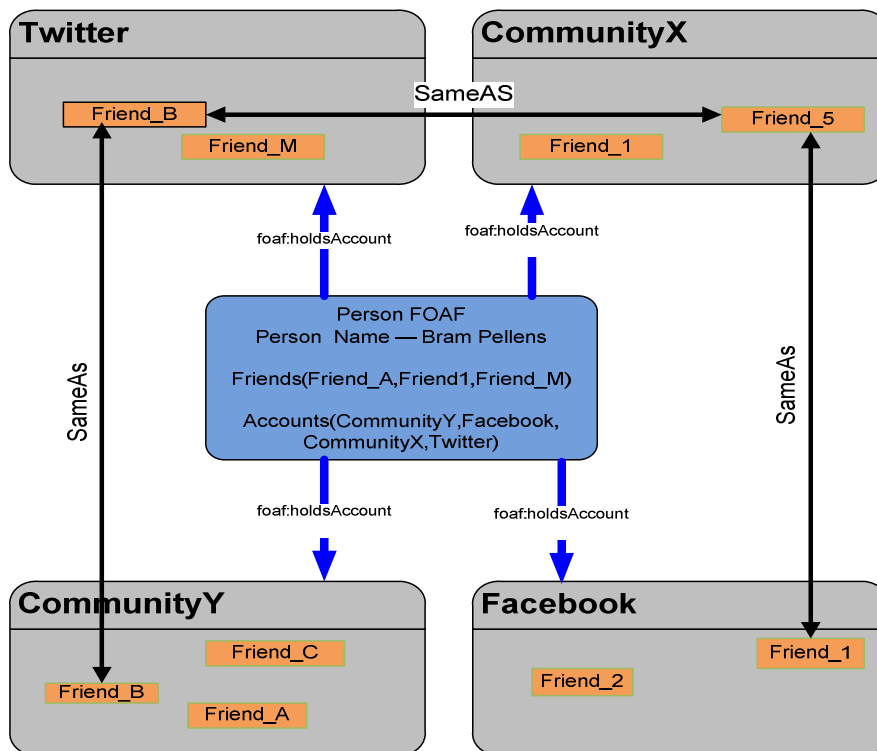


Figure 3-5 : FOAF Mining

3.1.2 SIOC Mining

As seen in Figure 3-1, contributions in the form of posts “Post1” and “Post2” were made by the person in CommunityX whereas “PostN” and “PostM” were made in CommunityY. Figure 3-6 shows the concept of posts in the different online communities.

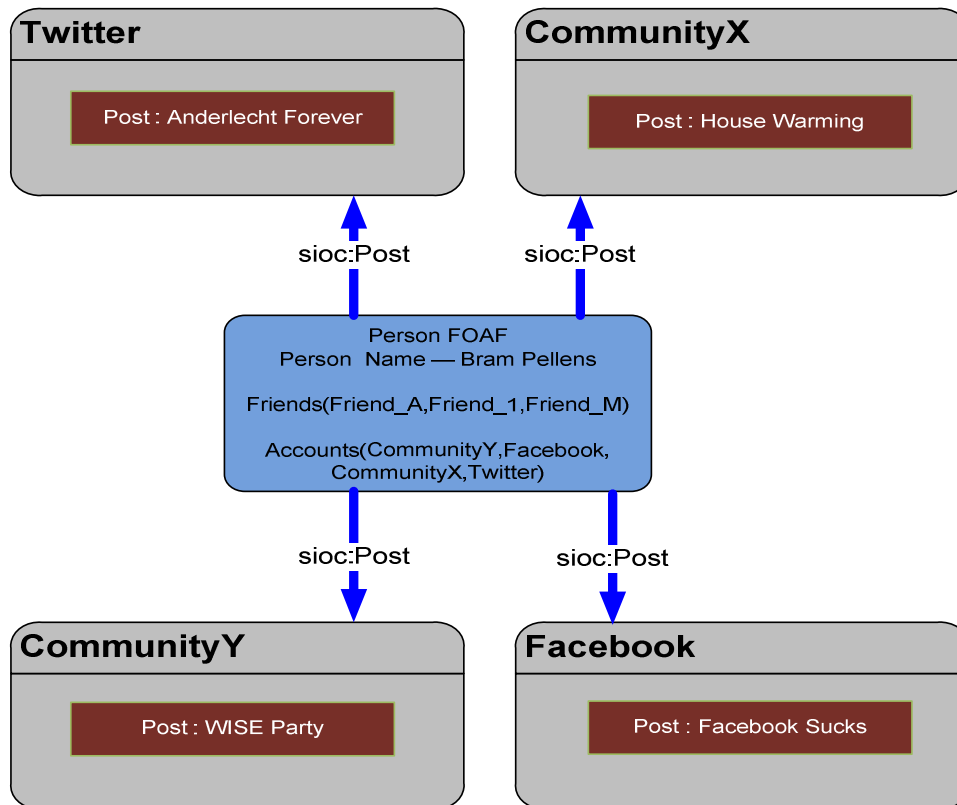


Figure 3-6 : SIOC Mining

In order to extract data from these communities, the communities need to have a SIOC enabled exporter. Currently communities such as those based on WordPress, b2evolution, Drupal and other software³⁸ have extended scripts that can be installed to enable exportation of RDF data in the SIOC format. With the exporter enabled, RDF format data as shown in Figure 3-7 could be picked from a 2D online community such as a WordPress blog, a Microblogging service or a social network. In this case Figure 3-7 shows an extract of the “Post:WISE Party” that we see under CommunityY in Figure 3-6. In this RDF data we can see the SIOC vocabulary used such as “sioc:Post” and “sioc:has_creator”. We realize that the person who is the creator of this post is represented using FOAF vocabulary “foaf:Person”. This reaffirms our discussion in Chapter 2 where we stated that SIOC borrows from other ontologies. Also the title of the post is represented with “dc:title” which is a vocabulary

³⁸ <http://sioc-project.org/applications>

borrowed for the Dublin Core Ontology. We also observe “dcterms:created” and sioc:content among other vocabulary used.

```

<sioc:Post rdf:about="http://CommunityY.info/wordpress/?p=38">
  <sioc:link rdf:resource="http://CommunityY.info/wordpress/?p=38"/>
  <sioc:has_container
rdf:resource="http://CommunityY.info/wordpress/index.php?sioc_type=site#weblog"/>
  <dc:title>WISE Party </dc:title>
  <sioc:has_creator>
    <sioc:User rdf:about="http://CommunityY.info/wordpress/?author=1"
rdf:label="admin">
      <rdfs:seeAlso
rdf:resource="http://CommunityY.info/wordpress/index.php?sioc_type=user&sioc_id=1"/
>
    </sioc:User>
  </sioc:has_creator>
  <foaf:makes
rdf:label="admin">
    <foaf:Person rdf:about="http://CommunityY.info/wordpress/?author=1#foaf"
rdf:label="admin">
      <foaf:mbox_sha1sum>e06b7bf3eb185115a527dc59e039ccc188ce00b3</foaf:mbox_sha
1sum>
      <foaf:homepage rdf:resource="http://CommunityY.info/foaf.html"/>
      <rdfs:seeAlso
rdf:resource="http://CommunityY.info/wordpress/index.php?sioc_type=user&sioc_id=1"/
>
    </foaf:Person>
  </foaf:makes>
  <dcterms:created>2010-01-03T16:40:20Z</dcterms:created>
  <sioc:content>
<p>Be Ready for the WISE Party that is dues 11th July : ALL are invited.</p>
</content:encoded>
  <sioc:topic rdf:label="Party"
rdf:resource="http://CommunityY.info/wordpress/?cat=3"/>
  <sioc:topic rdf:label="WISE"
rdf:resource="http://CommunityY.info/wordpress/?cat=1"/>
  <sioc:links_to rdf:resource="http://CommunityY.info/wordpress/wp-
content/uploads/2010/01/wise.jpg" />
</sioc:Post>

```

Figure 3-7: Sample Exported RDF

Figure 3-1 “SIOC Extract” shows that the SIOC data will be extracted from the different communities and stored in the “SIOC Data” file. Similar to FOAF mining a server technology script can be used to visit the account through HTTP and retrieve the RDF data. As seen in the Figure 3-1 Post1, Post2 and PostM were extracted and make up the SIOC data file.

3.1.3 Integration

As seen in Figure 3-1, at this stage we have two files, the FOAF Data and SIOC data file. We need to merge these files into one “Person Communities” file which is an instantiation of the Person Communities Ontology. In the Person Communities Ontology we express the concepts that represent the Person Profile, Friends List and Posts. This integration seen with the integration arrow in the Figure 3-1 is done using XSLT. Figure 3-8 shows the integration files to be used. The final result Person Communities Ontology will consist of the Person Profile, Friends List and Posts from the different communities.

File	Sample Extract	Comments
FOAFData.xml	<pre><FOAFDATA> <PersonProfile > <NAME>Bram</NAME> </PersonProfile> <Friends> <NAME>Frederic</NAME> </Friends> </FOAFDATA></pre>	Skeleton look of FOAF Data file. This is the file that was created at FOAF Mining stage.
SIOCData.xml	<pre><SIOCDATA> <PersonPosts> <Post> <posttitle>WISE Party</posttitle> </Post> </PersonPosts> </SIOCDATA></pre>	Skeleton look of SIOC Data file. This is the file that was created at the SIOC Mining Stage.
DataFilesList.xml	<pre><mergeData newRoot="Person_Communities_Data"> <fileList> <fileItem>FOAFData.xml</fileItem> <fileItem>SIOCData.xml</fileItem> </fileList> </mergeData></pre>	This file lists the two files we are going to merge.
Merge.xsl	<pre><xsl:template match="/"> <xsl:element name="{mergeData/@newRoot}"> <xsl:apply-templates select="mergeData/fileList/fileItem"/> </xsl:element> </xsl:template> <xsl:template match="fileItem"> <xsl:copy-of select="document(.)"/> </xsl:template></pre>	This xslt template is used to merge the two files.

PersonCommunities.xml	<pre> <Person_Communities_Data> <FOAFDATA> <PersonProfile> <NAME>Bram</NAME> </PersonProfile> <Friends> <NAME>Frederic</NAME> </Friends> </FOAFDATA><SIOC DATA> <PersonPosts> <Post> <posttitle>WISE Party</posttitle> </Post> </PersonPosts> </SIOC DATA></Person_Communities_Data> </pre>	This is the resulting Person Communities Ontology Extract. It contains the FOAF and SIOC Data.
-----------------------	---	--

Figure 3-8: Integration Files

3.2 3D Transformation

Having extracted the data and stored it in the Person Communities Ontology, the next step is the 3D transformation. In this transformation, XSLT will be used in combination with a server technology to generate X3D files. The use of XSLT enables concepts to be represented with different templates. At one stage any user can replace one template with another to achieve the desired 3D environment as discussed in the following sections.

3.2.1 Template Mapping

Different users may want to view their 3D environment with different representation for some concepts. Take for instance, one user may want a TV screen to view posts whereas another user may want a wall painting for the posts, the other user may want to view friends photos frames on the wall another may want a box to have these photos in. In order to fit the need of different users' preferences, we will use the concept of templates. Different templates can be used to represent the same concept (such as Post, Person) that we want to appear in the 3D Virtual World. As seen in Figure 3-1, there is a "Transformation Templates Repository" in which several templates can be stored and utilised when needed. Put differently, this repository can contain many templates (some user created) for friends, posts and profiles.

The templates discussed so far were more about individual virtual objects that will populate a Virtual World. But we need also template that can be used to model the Virtual World itself. In our approach, we have defined a "World XSL" template that is used to represent the

Virtual World. The Virtual World would be something like a Room or a Compound in which the different objects that will represent the concepts will be placed. This world sets the bounds of space the user will interact with. The World XSL will import the Friends XSL, Profile XSL and Posts XSL as seen in Figure 3-1. The imported XSL will be based on what the user will specify using a form on which the user will select his/her preferred templates.

The skeleton of the main XSLT file would look like the one in Figure 3-9 where `person_profile_template1.xml`, `person_post_template1.xml` and `person_friend_template1.xml` are templates that are being imported and applied using `apply-templates` function. Alternatively if the XSL templates were given names, a function like `xsl:call-template` could be used. It takes the name as a parameter.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:import href="person_profile_template1.xml"/>
<xsl:import href="person_post_template1.xml"/>
<xsl:import href="person_friend_template1.xml"/>

<xsl:template match="/" name="World_XSL">
  <xsl:apply-templates/>
</xsl:template>
</xsl:stylesheet>
```

Figure 3-9: Main XSL File Skeleton

It is important to note that the imported template may also import other XSL, e.g. a template for representing friends (`person_friend_template1.xml`) may import an XSL to make a list of box objects.

3.2.2 Code Generation

As seen in Chapter 2, using XSLT can be handy in the generation of XML formatted code such as X3D which is the required format in this approach. The different concepts can be realized with different Proto objects VR building blocks that are coming from the Object Library as seen in Figure 3-1. For instance friends can be represented with an XSLT template that uses Proto Box to represent each friend and Posts can be represented with an XSLT template that uses a Cylinder to represent a surface on which posts are written in a text format. After all the different required XSLT templates are put together, the code for the 3D environment can be automatically generated by adding the different templates to the main X3D file as inline

nodes. The inline node will be provided with the URL of the X3D code to import. An extract from the main X3D code file would look like the line below where different inline files have been added.

```
<Scene>  
  <Inline url=" ../X3D/Inline_Friends.x3d" " ../X3D/Inline_Posts.x3d" " ../X3D/Inline_Profile.x3d" />  
</Scene>
```

The code will have to be stored in a new file that will be created using a server technology such as PHP. At this point, the 3D file can be visualized in a 3D enabled web browser.

3.3 Adopting the Approach

Using the “FOAF and SIOC Based 3D Approach” offers a way for companies or individuals to generate rich content 3D Virtual Worlds. It also opens up means to capture data that can be used for some other Mashup. In order to use the approach, companies and individuals need to adapt their websites to the era of semantic web. Our approach starts with FOAF profile of an individual and then data like posts is extracted from communities that support SIOC RDF data export. We covered FOAF and SIOC in Chapter 2 under the ontology layer of the semantic web framework. In order for websites of individuals or companies to be able to support this kind of data export, they should have RDF data added based on SIOC ontology. Individuals who would like to use the approach will need a basic FOAF file that can be created with an online tool³⁹.

³⁹ <http://www.ldodds.com/foaf/foaf-a-matic>

4 Case Study

This section looks at how the proposed “FOAF and SIOC Based 3D Approach” described in Chapter 3 has been applied to transform content from 2D online communities into a 3D Virtual World. The Approach follows the steps below:

Phase 1: Social Media Mining

1. The user provides a FOAF profile.
2. FOAF Mining: Here we extract FOAF profile data and store it in a new “FOAF Data” XML file.
3. SIOC Mining: Here we extract SIOC based data from online communities and store it in a “SIOC Data” XML file.
4. Integration: The FOAF Data and SIOC Data XML files are integrated and stored in the Person Communities Ontology file.

Phase 2: 3D Transformation

5. Template Mapping: The user chooses XSL templates that will be used to represent the domain concepts. These user chosen templates are imported into the main world template.
6. Code Generation: XLST is used to generate X3D code where different XSL templates can create files that are imported as inline X3D nodes.

In order to start off the process some preparatory work as discussed in the following section was required.

4.1 Preparatory Steps

4.1.1 Set up of Online Communities

Two online communities based on Wordpress 2.9 were setup. These communities are accessible on <http://johnmark.info/wordpress/> and <http://hipipo.com/sports/>. A WordPress SIOC Exporter⁴⁰ script was added as a plugin to these communities. In order to easily recognize the availability of RFD data in these communities, Semantic Radar⁴¹ a semantic

⁴⁰ <http://sw.deri.org/svn/sw/2005/08/sioc/wordpress/>

⁴¹ <http://sioc-project.org/firefox>

metadata detector for Mozilla Firefox was installed on the test computer. Semantic Radar is a browser extension which inspects web pages for links to Semantic Web metadata and informs about presence of it by showing an icon in the status bar of the browser. Currently it supports RDF auto discovery (SIOC, FOAF, DOAP and other type) and RDFa metadata detection. As shown in Figure 4-1, the semantic radar icon display indicates the existence of RDF data based on the SIOC ontology.



Figure 4-1 : WordPress with SIOC RDF data under Semantic Radar

Semantic Radar Icon in Firefox browser represents the existence of SIOC RDF data on <http://johnmark.info/wordpress/>

4.1.2 Creation of FOAF Profile

As discussed in Chapter 3, the FOAF profile will be the starting point in the process to transform the data from the 2D online communities to a 3D Virtual World. This FOAF profile will contain the basic person properties, list of friends and the list of accounts a person holds on different communities. Figure 4-2 is an extract from the FOAF profile that has been

created for the purpose of testing this approach. This is a profile of a person called “John Mark”. In the extract (See Figure 4-2) we see nodes like foaf:name used to represent John’s name, foaf:holdsAccount used to list John’s online accounts and foaf:knows used to list John’s friends.

```

<?xml version="1.0"?>
<rdf:RDF
  <foaf:PersonalProfileDocument rdf:about="">
    <dc:title>Ssebunnya's FOAF</dc:title>
    <dc:description>Friend-of-a-Friend file for Ssebunnya</dc:description>
    <foaf:maker rdf:resource="#me"/>
    <foaf:primaryTopic rdf:resource="#me"/>
    <foaf:topic rdf:resource="#me"/>
    <terms:created>2010-02-03T11:12:45-05:00</terms:created>
    <terms:modified>2010-02-03T11:12:45-05:00</terms:modified>
  </foaf:PersonalProfileDocument>

  <foaf:Person rdf:ID="me">
    <foaf:name>John</foaf:name>
    <foaf:givenname>Mark</foaf:givenname>
    <foaf:family_name>John</foaf:family_name>
    <foaf:birthday>September 2</foaf:birthday>
    <foaf:locality>Brussels</foaf:locality>
    <foaf:region>Brussels</foaf:region>
    <foaf:country>Belgium</foaf:country>
    <foaf:gender>male</foaf:gender>
    <foaf:holdsAccount>
      <foaf:OnlineAccount>
        <foaf:accountServiceHomepage rdf:resource="http://johnmark.info/wordpress/" />
        <foaf:accountProfilePage rdf:resource="http://johnmark.info/wordpress/?author=1" />
        <foaf:accountName>Admin</foaf:accountName>
      </foaf:OnlineAccount>
      <foaf:OnlineAccount>
        <foaf:accountServiceHomepage rdf:resource="http://hipipo.com/sports/" />
        <foaf:accountProfilePage rdf:resource="http://hipipo.com/sports/?author=4" />
        <foaf:accountName>nico</foaf:accountName>
      </foaf:OnlineAccount>
    </foaf:holdsAccount>
    </foaf:holdsAccount>
    <foaf:knows>
      <foaf:Person>
        <foaf:name>Frederick</foaf:name>
        <foaf:holdsAccount>
          <foaf:OnlineAccount>
            <foaf:accountServiceHomepage rdf:resource="http://www.facebook.com/" />
            <foaf:accountProfilePage rdf:resource="http://www.facebook.com/profile.php?id=37007158" />
            <foaf:accountName>Frederick </foaf:accountName>
          </foaf:OnlineAccount>
        </foaf:holdsAccount>
      </foaf:Person>
      <foaf:Person>
        <foaf:name>Hadi</foaf:name>
        <foaf:holdsAccount>
          <foaf:OnlineAccount>
            <foaf:accountServiceHomepage rdf:resource="http://www.facebook.com/" />
            <foaf:accountProfilePage rdf:resource="http://www.facebook.com/profile.php?id=505932754" />
            <foaf:accountName>Hadi Jallow</foaf:accountName>
          </foaf:OnlineAccount>
        </foaf:holdsAccount>
      </foaf:Person>
    </foaf:knows>
  </foaf:Person>
</rdf:RDF>

```

Figure 4-2 : Created FOAF Profile Extract

4.2 Using the Approach

From the discussion in the previous section, the preparatory work has been done and now it is time to use the approach to generate 3D Virtual World.

4.2.1 Phase 1: Social Media Mining

4.2.1.1 Provide FOAF Profile

To start off the FOAF mining, the user is provided with the interface shown in Figure 4-3. The interface is used to provide a link to the FOAF profile and when ready the user will click on the search button.

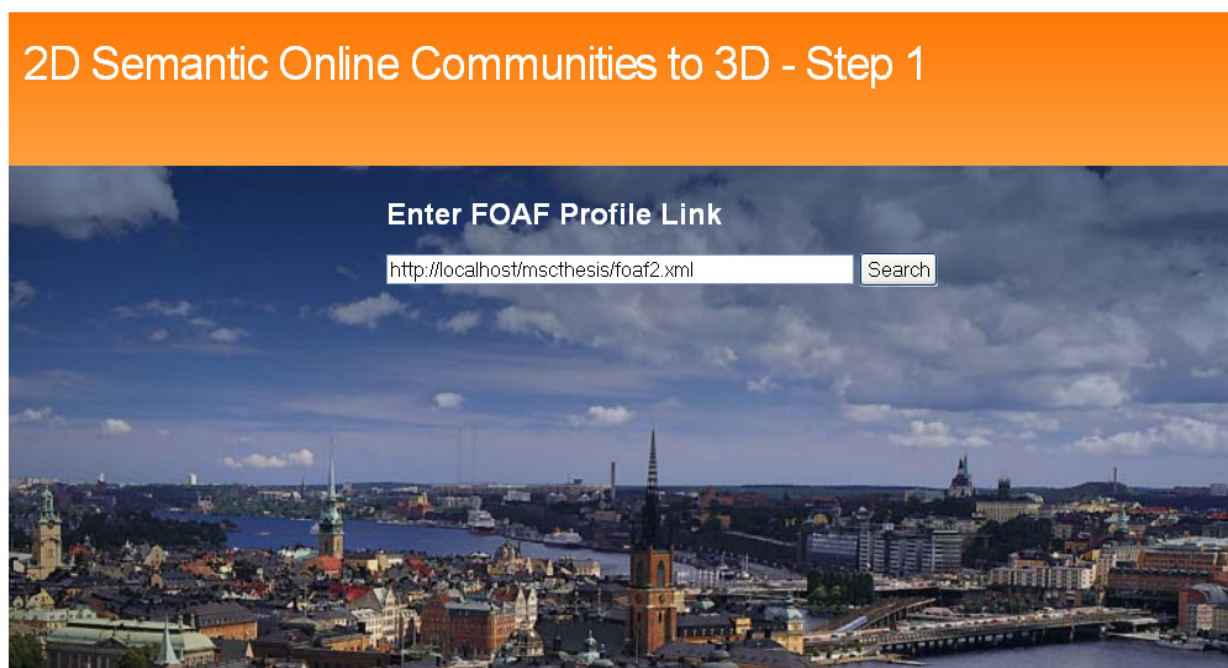


Figure 4-3 : FOAF Link Form

4.2.1.2 FOAF Mining: Extract FOAF Data

Once the profile link is provided the following occurs

1. Check if input is a valid URL.
2. Using the defined HTTP protocol, visit the provided URL, retrieve the remote file and store its content in temporary memory.
3. Check if it is a valid XML file.
4. Load an XSL template that will be used to transform this file to an XHTML format that we shall show to the user.
5. Check if the XSL is a valid XSL.

6. If the above are performed correctly, we proceed to pass the XML and XSL files to a transform function that uses PHP DOM functions represented in the code snippet in Figure 4-4. Otherwise, a failure error will be returned.

```
function transformDocument_domxml5($xml, $xsl, $params) {
    $magic_quotes_runtime_orig = get_magic_quotes_runtime();
    set_magic_quotes_runtime(0);
    $xmlDom = new DOMDocument;
    $xslDom = new DOMDocument;
    $xmlDom->loadXML($xml);
    $xslDom->loadXML($xsl);
    $proc = new XSLTProcessor;
    foreach ($params as $key => $value) {
        $proc->setParameter("", $key, $value);
    }
    $old_error_reporting = error_reporting(E_ALL);
    $old_error_handler = set_error_handler(array(&$this, 'util_errorHandler'));
    $proc->importStyleSheet($xslDom);
    $result = $proc->transformToDoc($xmlDom);
    restore_error_handler();
    error_reporting($old_error_reporting);
    if ($this->hasError()) {
        $myError = $this->getErrorFromCode('MM_TRANSFORMATION_ERROR');
        $myError .= $this->getRawError();
    }
    $myError .= $this->getErrorFromCode('MM_TRANSFORM_D5_ERROR', array($this->xslname));
    $myError .= $this->util_htmlentities($xsl);
    $this->setError($myError);
    set_magic_quotes_runtime($magic_quotes_runtime_orig);
    return;
}
$output = $result->saveXML();
set_magic_quotes_runtime($magic_quotes_runtime_orig);
return $output;
}
```

Figure 4-4 : PHP and XSL Snippet to Extract FOAF Profile

In the code snippet in Figure 4-4, the FOAF file and the XSL templates are loaded as an XML file using the loadXML() function that is called with a DOMDocument⁴² object. A new XSLTProcessor⁴³ object is created. With this object, we can have access to different XSLTProcessor functions that will be used to traverse and transform the XML document to another document in this case XHTML using the rules that were specified in the XSL template. After this transformation is done the new document is saved in memory as a valid XML using saveXML() function. This document is then returned and displayed on the screen as shown in Figure 4-5.

⁴² <http://php.net/manual/en/class.domdocument.php>

⁴³ <http://php.net/manual/en/class.xsltprocessor.php>

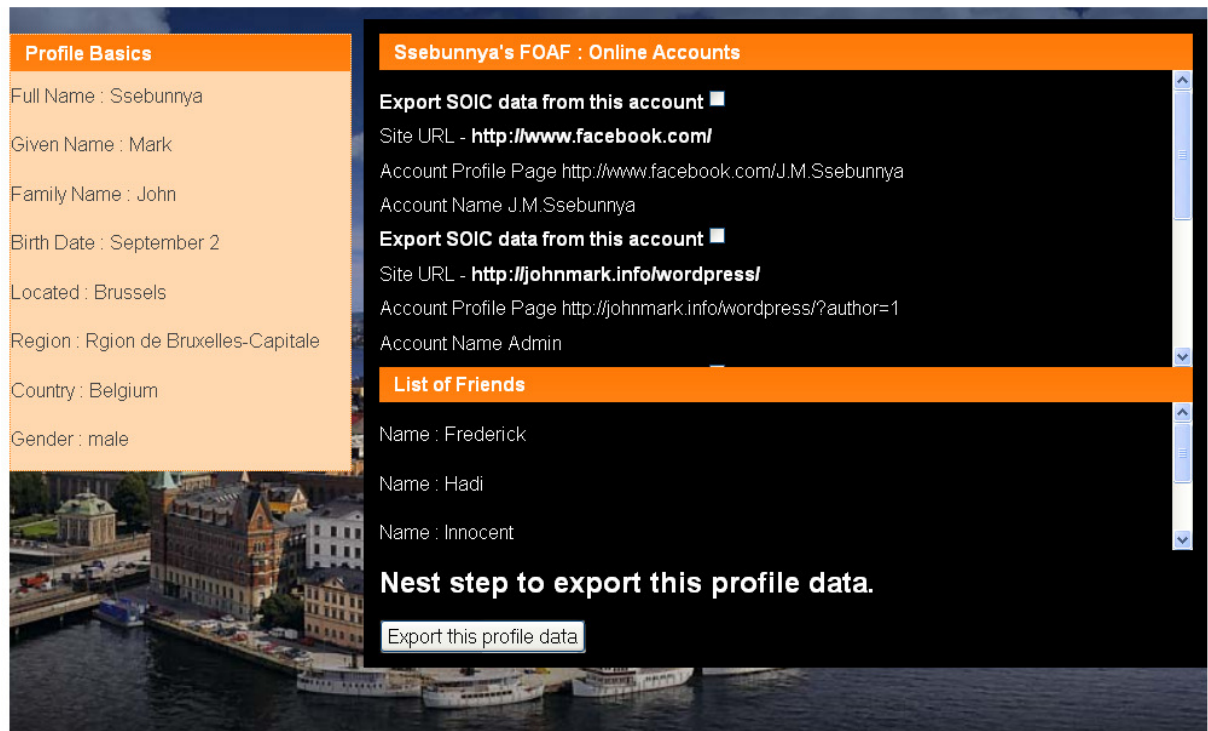


Figure 4-5 : Preview FOAF Data

The purpose of presenting this data to the user is to affirm that this is actually the right data. Once the user has seen the data and is ready to proceed, a button is available to export this data and store it in the FOAF Data XML file. Figure 4-6 shows the extracted data of “John Mark” (The user we introduced earlier). In it we see his basic profile details, the online community accounts he holds on johnmark.info, facebook.com, twitter.com and hipipo.com. We also see a list of his friends namely Frederick, Hadi, Innocent and Betty. Appendix A: PHP Snippet shows a code snippet that is used to extract the data, create a well formed XML file and store the new file. This is done largely with the use of PHP DOM⁴⁴ Module functions.

```
<?xml version="1.0"?>
<foaf_data version="foaf_basic">
  <Person>
    <name>John</name>
    <givenname>Mark</givenname>
    <family_name>Mark</family_name>
    <birthday>September 2</birthday>
    <locality>Brussels</locality>
    <region>Brussels</region>
    <country>Belgium</country>
    <gender>male</gender>
    <holdsAccount>
      <OnlineAccount>
        <accountServiceHomepage>http://www.facebook.com/</accountServiceHomepage>
        <accountProfilePage>http://www.facebook.com/J.M.Ssebunnya</accountProfilePage>
        <accountName>J.M.Ssebunnya</accountName>
```

⁴⁴ <http://php.net/manual/en/class.domdocument.php>

```

</OnlineAccount>
<OnlineAccount>
  <accountServiceHomepage>http://johnmark.info/wordpress/</accountServiceHomepage>
  <accountProfilePage>http://johnmark.info/wordpress/?author=1</accountProfilePage>
  <accountName>Admin</accountName>
</OnlineAccount>
<OnlineAccount>
  <accountServiceHomepage>http://hipipo.com/sports/</accountServiceHomepage>
  <accountProfilePage>http://hipipo.com/sports/?author=4</accountProfilePage>
  <accountName>nico</accountName>
</OnlineAccount>
<OnlineAccount>
  <accountServiceHomepage>http://twitter.com/</accountServiceHomepage>
  <accountProfilePage>http://twitter.com/jmssebunnya</accountProfilePage>
  <accountName>jmssebunnya</accountName>
</OnlineAccount>
</holdsAccount>
<knows>
  <Person>
    <name>Frederick</name>
  </Person>
  <Person>
    <name>Hadi</name>
  </Person>
  <Person>
    <name>Innocent </name>
  </Person>
  <Person>
    <name>Betty </name>
  </Person>
</knows>
</Person>
</foaf_data>

```

Figure 4-6 : Extracted FOAF Data

We discussed in Chapter 3 that FOAF mining can proceed with visiting online accounts that allow FOAF export from which we can extract other FOAF files and merge them together into one FOAF data file. We noted that in case some friends appear in several communities FOAF data, we may be faced with a problem of duplication in our friends list. We hence would have to sort out the duplicates from the list manually. In this case study, our person John Mark holds accounts with johnmark.info, facebook.com, twitter.com and hipipo.com. With the current state of the art, none of these online communities allow FOAF profile export in an open way, though we are optimistic that in the future this would be possible with social networks such as Facebook. Though at this stage we not able to extract FOAF data from these communities such as Facebook, it is important to remember that we have FOAF data that has been retrieved from the user provided FOAF file.

4.2.1.3 SIOC Mining: Extract SIOC Data

Having captured the person profile data and taken a list of the online communities, the user will be able to proceed to the next step in which data will be exported from communities

that have enabled the SIOC exporter. In the case study the two communities stated earlier are WordPress communities accessible on <http://johnmark.info/wordpress/> and <http://hipipo.com/sports/>. A user will have an interface as shown in Figure 4-7 on which his/her communities are listed and will proceed to use a button "Export SIOC Data" to start the process of extracting the SIOC data from these communities.

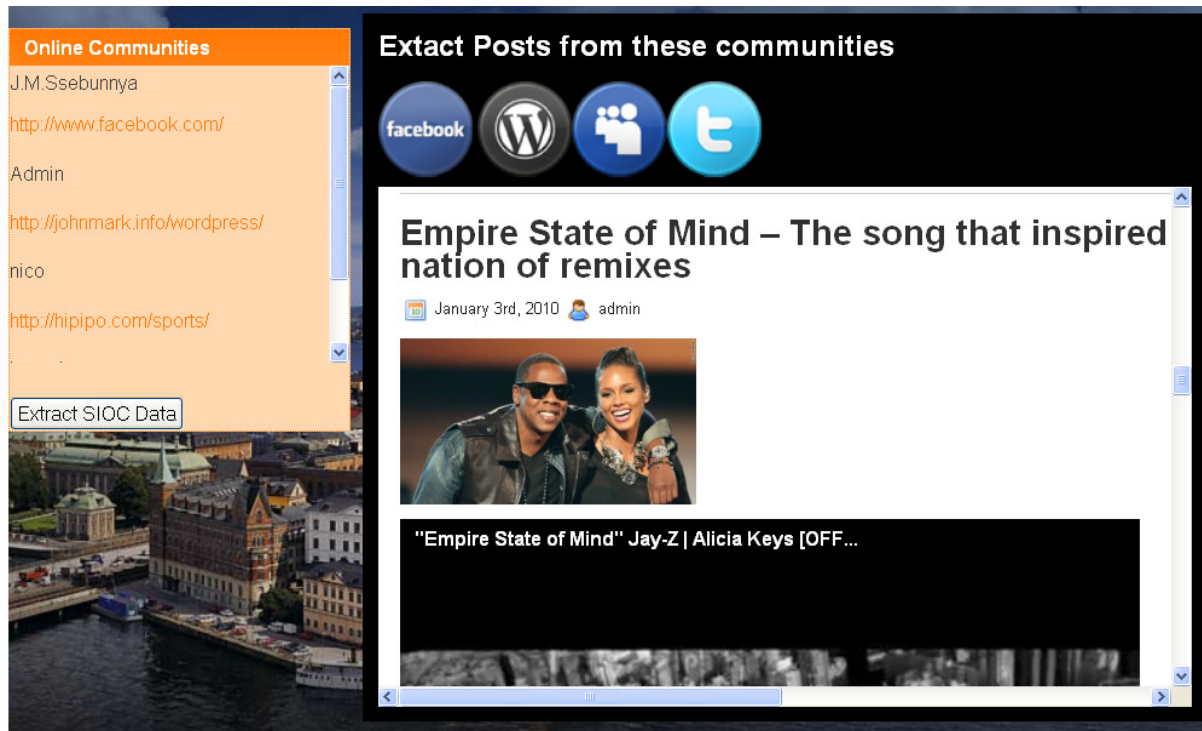


Figure 4-7 : List User's 2D Online Communities

In this case study when one of the communities accessible on <http://johnmark.info/wordpress> is visited, we read the RDF extract as seen in Figure 4-8. This extract shows the `sioc:Site` address, `sioc:Forum` and under each `sioc:Container` we find a particular `sioc:Post` in which we can see `dc:title` (the post title expressed using Dublin Core) and a link to the post details. We visit each of these posts individually to get the post details.

```
<sioc:Site rdf:about="http://johnmark.info/wordpress/">
  <dc:title>Celeb Space</dc:title>
  <dc:description>Weblog: Celeb Space</dc:description>
  <sioc:link rdf:resource="http://johnmark.info/wordpress/" />
  <sioc:host_of rdf:resource="http://johnmark.info/wordpress/index.php?sioc_type=site#weblog/" />
  <sioc:has_group rdf:resource="http://johnmark.info/wordpress/index.php?sioc_type=site#authors" />
</sioc:Site>

<sioc:Forum rdf:about="http://johnmark.info/wordpress/index.php?sioc_type=site#weblog">
  <sioc:name>Main blog at Celeb Space</sioc:name>
  <sioc:link rdf:resource="http://johnmark.info/wordpress/index.php?sioc_type=site#weblog/" />
  <sioc:container_of>
    <sioc:Post rdf:about="http://johnmark.info/wordpress/?p=49" dc:title="Taylor Swift, Beyonce
```

```

grab Grammys, but Jackson kids steal show">
    <rdfs:seeAlso
rdf:resource="http://johnmark.info/wordpress/index.php?sioc_type=post&sioc_id=49"/>
    </sioc:Post>
</sioc:container_of>
<sioc:container_of>
    <sioc:Post rdf:about="http://johnmark.info/wordpress/?p=38" dc:title="Empire State of Mind
&#8211; The song that inspired a nation of remixes">
        <rdfs:seeAlso
rdf:resource="http://johnmark.info/wordpress/index.php?sioc_type=post&sioc_id=38"/>
        </sioc:Post>
    </sioc:container_of>
<sioc:container_of>
    <sioc:Post rdf:about="http://johnmark.info/wordpress/?p=28" dc:title="Britney Spears Ranks
2009&#8217;s Most Ridiculous Britney Stories">
        <rdfs:seeAlso
rdf:resource="http://johnmark.info/wordpress/index.php?sioc_type=post&sioc_id=28"/>
        </sioc:Post>
    </sioc:container_of>
<sioc:container_of>
    <sioc:Post rdf:about="http://johnmark.info/wordpress/?p=22" dc:title="Beyonc , Taylor Swift,
Black Eyed Peas to Perform at Grammys">
        <rdfs:seeAlso
rdf:resource="http://johnmark.info/wordpress/index.php?sioc_type=post&sioc_id=22"/>
        </sioc:Post>
    </sioc:container_of>
</sioc:Forum>

```

Figure 4-8 : Sample SIOC Site RDF

Using an example of the first post title “Taylor Swift, Beyonce grabs Grammys, but Jackson kids steal show”, we visit its RDF link at which we read its details in RDF format shown in Figure 4-9. Among other XML nodes in the extract, we can see the post title represented using dc:title, the creator of the post represented with sioc:has_creator and the post details under sioc:content. We would then have to go through similar steps like in section 4.2.1.1 using XSLT and PHP DOM functions in order to retrieve the valid XML file and store the content in the SIOC data file.

```

<foaf:Document rdf:about="">
    <dc:title>SIOC Post profile for "Celeb Space"</dc:title>
    <dc:description>A SIOC profile describes the structure and contents of a weblog in a machine readable form.
For more information please refer to http://sioc-project.org/.</dc:description>
    <foaf:primaryTopic rdf:resource="http://johnmark.info/wordpress/?p=49"/>
    <admin:generatorAgent rdf:resource="http://rdfs.org/sioc/wp-sioc.php?version=1.25"/>
</foaf:Document>

<sioc:Post rdf:about="http://johnmark.info/wordpress/?p=49">
    <sioc:link rdf:resource="http://johnmark.info/wordpress/?p=49"/>
    <sioc:has_container rdf:resource="http://johnmark.info/wordpress/index.php?sioc_type=site#weblog"/>
    <dc:title>Taylor Swift, Beyonce grab Grammys, but Jackson kids steal show</dc:title>
    <sioc:has_creator>
        <sioc:User rdf:about="http://johnmark.info/wordpress/?author=2" rdfs:label="johnmarc_mj">
            <rdfs:seeAlso
rdf:resource="http://johnmark.info/wordpress/index.php?sioc_type=user&sioc_id=2"/>
            </sioc:User>

```

```

</sioc:has_creator>
<foaf:maker>
  <foaf:Person rdf:about="http://johnmark.info/wordpress/?author=2#foaf"
rdf:label="johnmarc_mj">

    <foaf:mbox_sha1sum>5993ca791f86f10be6da6f4adf0e0682da66953b</foaf:mbox_sha1sum>
      <foaf:homepage rdf:resource="http://hipipo.com"/>
      <rdfs:seeAlso
rdf:resource="http://johnmark.info/wordpress/index.php?sioc_type=user&sioc_id=2"/>
      </foaf:Person>
    </foaf:maker>

    <dcterms:created>2010-02-10T14:14:25Z</dcterms:created>
    <sioc:content>Los Angeles, California (CNN) -- The most memorable moment of the 52nd annual Grammy
Awards on Sunday may be the onstage appearance of Michael Jackson's two eldest children to accept their father's
lifetime achievement award.
</sioc:content>
    <content:encoded><![CDATA[<p>Los Angeles, California (CNN) &#8212; The most memorable moment of
the 52nd annual Grammy Awards on Sunday may be the onstage appearance of Michael Jackson&#8217;s two eldest
children to accept their father&#8217;s lifetime achievement award.</p>
]]></content:encoded>
    <sioc:topic rdf:label="CelebsNews" rdf:resource="http://johnmark.info/wordpress/?cat=3"/>
</sioc:Post>

```

Figure 4-9 : Sample Post RDF Extract

4.2.1.4 Integration: Person Communities Ontology

At this stage we have two XML files, FOAF Data and SIOC Data. We proceed to merge the two files into one Person Communities Ontology. This is still done with XSLT and PHP DOM functions. Figure 3-8 showed a detailed procedure of how the integration happens.

4.2.2 Phase 2: 3D Transformation

Having captured the required data and stored it in the Person Communities XML file, we can go ahead to transform this data to a X3D Virtual World. XSLT as discussed in Chapter 2 is fundamental in transformations from one XML format to another. X3D is an XML based language thus use of XSLT transformation will be handy in the generation of X3D code.

4.2.2.1 Template Mapping: Choose XSL Templates

Templates can be used to achieve a particular block of functionality in XSLT. Templates have been defined to represent different concepts that we want to have in the 3D Virtual World. Take for instance a screen can be used to write posts on it, display photos, videos etc. An X3D primitive object like a box can be used to represent this screen in the template. Figure 4-10 shows sample items that have been used to represent concepts in a template.

Concept	To be Map To	Primitive Proto Objects
Person Profile	Writing on Wall	Box, Text
Friends	List Photo Frame on Wall	Box or Sphere
Posts	TV Screen in sitting room	Text , Box

Figure 4-10 : Concepts Mapped to Primitive Objects

At this stage the user is provided with an interface shown in Figure 4-11 from which to select templates he/she wants to use to represent the concepts in a 3D Environment. After selecting the template, the user will click on “Transform to 3D” button.

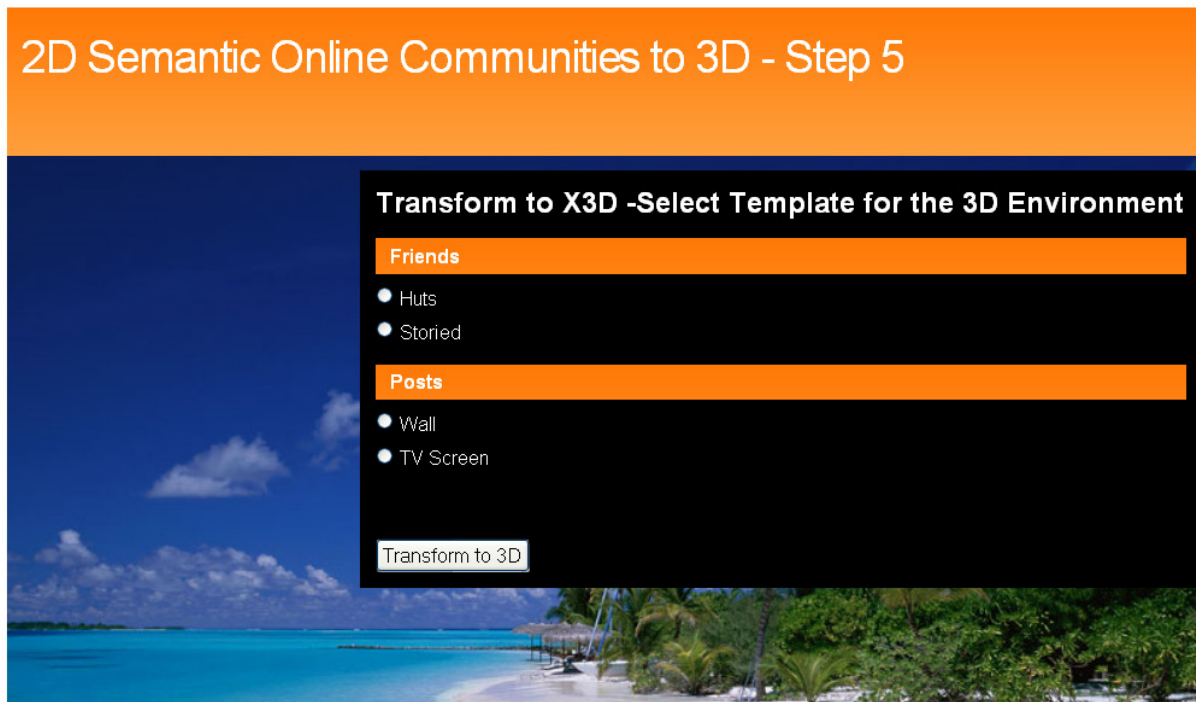


Figure 4-11 : Choosing Templates

4.2.2.2 Code Generation: Transform to 3D Environment

From the previous stage, after the user has clicked on “Transform to 3D”, the different specified templates such as shown in the code snippet in Figure 4-12 will be imported from a pool of templates. These templates will be applied in the main conversion template. Using a similar function as in the code snippet in Figure 4-4, the Person Communities file and the Main World XSL templates are passed as parameters, they are loaded as XML files and transformation is done using the XSLTProcessor. After this transformation is done, new documents are saved for the main X3D file and the other X3D files that are included in the main file are done so as inline nodes. Once the transformation is complete, the resulting 3D environment is visualized in a browser.

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="xml" encoding="utf-8" indent="yes"/>
<xsl:template match="foaf_sioc_combined/Person/knows" mode="huts" >
<xsl:for-each select="Person">
  <xsl:variable name="count">
```

```

<xsl:number/>
</xsl:variable>
<Group>
  <Transform scale='.5 1 .5' translation='{ -43 + $count + $count + $count + $count + $count +
$count } 24 20'>
    <Group USE='HUT' />
      <Transform translation='{ -2 + $count } 10 5'>
        <Shape>
          <Text string="{name}">
            <FontStyle family=""SANS"" horizontal='false' justify=""MIDDLE" "MIDDLE"" size='3.0'
style='BOLD' />
          </Text>
          <Appearance>
            <Material diffuseColor='0.8 0.1 0.4' />
          </Appearance>
        </Shape>
      </Transform>
    </Transform>
  </Group>
</xsl:for-each>
</xsl:template>
</xsl:stylesheet>

```

Figure 4-12 : Sample XSLT Template

5 Integration with Existing Approaches

The “FOAF and SIOC Based 3D Approach” discussed in Chapter 3 and elaborated with a case study in Chapter 4 uses different templates, which are manually created and assembled together by using XSLT transformation. In this section, we describe a few possible ways in which the existing approaches such as “Virtual Reality – With Intuitive Specifications Enabled” (VR-WISE) and “Conceptual Design Pattern Approach” (CoDePA) can be integrated with our approach or how our approach can facilitate those other approaches. We firstly discuss the VR-WISE approach as well as the CoDePA approach. Afterwards we go into more detail on how the VR-WISE Approach can be used to create templates that could be used to give a more appealing look to our 3D Virtual World. We also discuss how the concept of using XSL templates in the “FOAF and SIOC Based 3D Approach” can be integrated in the VR-WISE Approach in order to generate better templates using OntoWorld the VR-WISE support tool.

5.1 VR-WISE Approach

The VR WISE approach uses ontologies to facilitate the development of Virtual Worlds. The so called Domain Ontology can be considered as representation of a domain conceptualization describing possible concepts and relationships between these concepts. The information contained in such domain ontology is well suited to be used as a starting point to specify a Virtual World at a conceptual level. This can be done by allowing the designer to select concepts from the domain ontology that are needed in the Virtual World and by creating properties and relationships between those concepts from the domain ontology (Wesley, et al., 2004). In the following section, we discuss the stages to develop Virtual Worlds using the VR-WISE Approach shown in Figure 5-1 (Pellens, et al., 2007).

5.1.1 Specification Stage

In the specification stage, the designer specifies his VR application at a conceptual level using the domain terminology which may combine more than one domain ontology from which to select the domain concepts that are needed in the Virtual World. VR modeling concepts are defined and described in the VR Conceptual Modeling Ontology. The Virtual Reality Conceptual Modeling Ontology describes the high-level conceptual modeling concepts that a non-VR-specialist can use to specify the Virtual World in an intuitive way. A concept such as

spatial relation (e.g. 'InFrontOf', 'Behind', 'LeftOf', 'InsideOf') is used to place objects in the World. The concepts allow describing complex objects, to specify behavior and to express interaction.

The Domain Ontology describes, at a conceptual level, the concepts available in the domain together with the properties and relations between them. The World Specification does the actual description of the Virtual World at a conceptual level by instantiating concepts from the Domain Ontology and describing how they are related, where they are placed in the World, how to interact with them and what behavior they have.

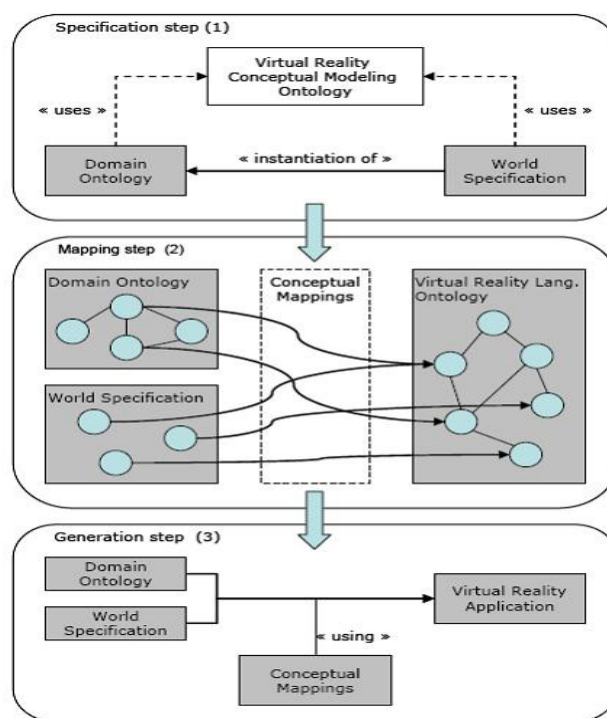


Figure 5-1 : VR-WISE Approach

5.1.2 Mapping Stage

After the specification stage, the next thing to do is specifying how the domain concepts and the instances will be represented in the Virtual World. This is done by mapping the concepts defined in the Domain Ontology and the instances defined in the World Specification onto VR building blocks. By VR building blocks we mean VR primitives (e.g. sphere, box), constraints and so on. In this stage, also a number of (internal) ontologies and mappings are used such as the VRML Ontology which defines the concepts available in VRML, like the primitive object types (e.g. sphere, box, etc.). The Meta Mapping is used to describe how the

tool keeps track of the mapping of the OWL concepts onto the X3D concepts. The Domain Mapping is an instantiation of the Meta Mapping which defines a default representation of the domain concepts in the Virtual World. In instances where properties appearance will differ a bit from the one used by default, the World Mapping can overwrite the default mapping specified for a concept in the Domain Mapping.

5.1.3 Generation Stage

In the last stage, the code generation, the OntoWorld tool transforms information from the Domain Ontology, the World Specification and both mappings into a working Virtual World by generating X3D code. Following the steps described above, a virtual e-shop shown in Figure 5-2 was generated.



Figure 5-2 : VR-WISE Approach - Virtual Shop

5.2 CoDePA

The development of dynamic X3D/VRML Virtual Worlds is a complicated task which requires high technical skills and dedication of time. In order to simplify the development process for 3D designers, Pellens, et al. introduced CoDePA a Conceptual Design Pattern Approach to model Behavior for X3D worlds. CoDePA is based on generative design patterns in combination with a graphical notation. In CoDePA, behavior can be specified by instantiating patterns and customizing them to a particular context. More complex behavior can be constructed by combining different patterns. Furthermore, the approach allows more experienced designers to specify new behavior patterns and make them available to others

in a graphical way and ready to use. The concept of patterns is known from the Software Engineering domain where a design pattern specifies a well-defined solution to a problem that often appears when designing and developing software and allows capturing the expertise of others in a well-defined way. CoDePA uses Generative Design Patterns (O'Connell, 1995) which are similar to regular design patterns except that they are defined in such a way that it becomes possible to automatically generate code from them. They remove the burden of implementing the patterns from the user. In CoDePA this concept of Generative Design Patterns has been extended into what they call Visual Generative Design Patterns by expressing patterns using a graphical notation and by integrating them into existing graphical behavior (animation) modeling language. In the following section we discuss the process of designing and using patterns under the CoDePA Pattern Specification Framework shown in Figure 5-3 (Pellens, et al., 2008).

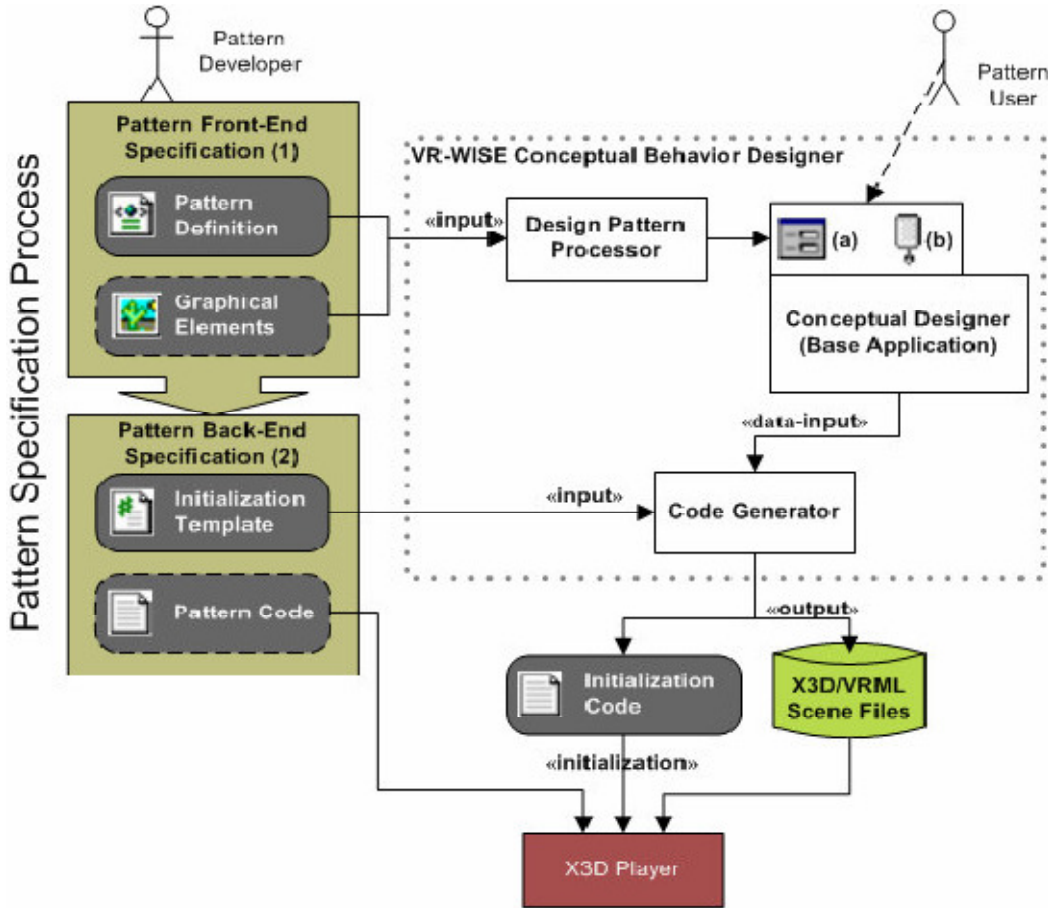


Figure 5-3 : CoDePA Pattern Specification Framework

Figure 5-3 also shows some of the most important modules of the VR-WISE Conceptual Behavior Designer in which the Design Pattern Processor processes some parts of the

pattern specification and the Code Generator uses the data from the designer together with some templates to generate code.

5.2.1 Specifying Patterns

CoDePA enables building new patterns and adding them to a pattern library from where they can be picked for future use. The process of specifying new patterns is divided into two phases namely the front-end specification and the back-end specification as shown in Figure 5-3.

5.2.1.1 Pattern Front-End Specification

The front-end specification allows the designer to describe the part of the pattern that will be seen by the user of the pattern (a behavior developer). As seen in Figure 5-3, specifying the front-end of a pattern consists of two kinds of specifications namely the specification of the graphical elements needed in the pattern and the definition of the pattern itself i.e., how the different graphical elements can be combined in the pattern. In defining the pattern, the pattern designer specifies the elements that are needed, their role, how they are connected, and the parameters (with default values) that are required. The pattern description is captured in an XML format (defined by a DTD), which has the benefit of being tool independent. CoDePA uses a graphical tool based on Microsoft Visio (Biafore, 2007) to ease the description of a pattern.

5.2.1.2 Pattern Back-End Specification

Specifying the back-end of a pattern is needed in order to be able to generate programming code. This specification consists of the specification of the template used to generate the initialization code, and the actual underlying pattern code itself. The initialization code (see Figure 9) will initialize (i.e. invoke) the actual pattern code. The pattern code is the programming code that implements the behavior described by the pattern. The initialization code will be generated from the information that has been entered by the pattern user when instantiating and customizing a particular pattern and a so-called initialization template (which needs to be given by the pattern designer).

5.2.1.3 Pattern Library

The Pattern Library will be the storage of different available patterns that have been created by the Pattern Designer. In CoDePA the Pattern Library can contain a variety of pattern

categories of which most are behavioral patterns. New patterns may be discovered and can be added to the library at any time

5.2.2 Using Patterns

In order to use a pattern it should have been specified as discussed earlier. Applying a pattern for the specification of a particular Virtual World is called instantiation with consist of pattern selection and pattern customization. To instantiate a pattern, the designer should select the appropriate pattern from the available library of patterns. When a pattern is selected, the different graphical elements and connections that compose this pattern are automatically dropped on the drawing canvas. The designer can specify proper names for different components and customize the pattern. Customization can be giving proper values to parameters (overriding the default values provided for the pattern) or adapting the default number of components in the pattern.

5.3 Integration

Having discovered that the VR-WISE approach can generate Virtual Worlds using a domain and world ontology, we can deploy it to generate some templates that will be used in the templates repository (See Figure 3-1) of the FOAF and SIOC Based 3D Approach. However, to develop more visually appealing templates, we need to have an improved mechanism to address issues arising with static data objects such as friends lists that can have from one to thousands of friends. CoDePA which integrates with VR-WISE introduced a concept of patterns to model behavior for X3D worlds, this concept of patterns can be extended to model static objects. In this section we discuss how we can generate templates with VR-WISE, how we can integrate FOAF and SIOC Based 3D Approach with VR-WISE and how we can extend CoDePA and OntoWorld the VR-WISE support tool to generate better templates for static representations like friends lists.

5.3.1 Templates Generation

In the “Person Communities Ontology” shown in Figure 3-1, we express the concepts for Person Profiles, Friends List and Posts. We can use VR-WISE approach OntoWorld tool to model and generate templates for each of these concepts in addition to generating the “Main World” templates. Below we elaborate how we can go through the phases of VR-WISE approach to generate templates that may be used in “FOAF and SIOC Based 3D Approach”.

5.3.1.1 Specification Phase

Using the OntoWorld tool a designer adds the concepts of the Person Communities Ontology domain while specifying their relationships such as 'InFrontOf', 'Behind', 'LeftOf', 'InsideOf' that are used to place objects in the Virtual World.

5.3.1.1.1 Domain Specification

With OntoWorld, the designer creates an ontology by introducing each concepts with its properties and by setting relationships like subclass of. As OntoWorld does not allow yet to import existing ontology, we create it using the tool. Figure 5-4 shows an XML example of the project created with OntoWorld where we see a concept node name "Person" and having properties such as "givenname", "country" etc. We also see a node like <Mappings /> which will be used to represent mappings information.

```
<?xml version="1.0" encoding="utf-8"?>
<ConceptualGraph >
  <Nodes>
    <ConceptNode name="Person" group="" cls="OntoWorld.Core.Statics.ConceptNode">
      <Properties>
        <Property name="name" type="String" value="" />
        <Property name="givenname" type="String" value="" />
        <Property name="family_name" type="String" value="" />
        <Property name="birthday" type="String" value="" />
        <Property name="locality" type="String" value="" />
          <Property name="region" type="String" value="" />
          <Property name="country" type="String" value="" />
          <Property name="gender" type="String" value="" />
      </Properties>
      <Annotations />
      <AnnotationRules />
      <Target>
      </Target>
      <Mappings />
    </ConceptNode>
  </Nodes>
</ConceptualGraph>
```

Figure 5-4: Domain Specification Extract

5.3.1.1.2 World Specification

As earlier stated, the World Specification does the actual description of the Virtual World at a conceptual level by instantiating concepts from the Domain Ontology and describing how they are related, where they are placed in the World, how to interact with them and what behavior they have. Figure 5-5 shows the world specification stored in XML based on the domain specification. It contains the mapping (see next section), the position of the instance

and its relation with other instances. Here we see that an instance of the concept Person is created with the property name “Mark”. We see the detail of the properties of this person Mark filled in the property nodes that were first seen in the domain specification file.

```

<ConceptualGraph cls="OntoWorld.Core.Graph.ConceptualGraph">
  <Nodes>
    <InstanceNode name="Mark" type="Person" cls="OntoWorld.Core.Statics.InstanceNode">
      <Angle>180</Angle>
      <Axis>Above-Below</Axis>
      <Cartesian>False</Cartesian>
      <OrientationBySides>True</OrientationBySides>
      <Orientation X="0" Y="3.141593" Z="0" />
      <Position X="0" Y="0" Z="0" />
      <Unit>degrees</Unit>
      <Properties>
        <Property name="name" type="String" value="Ssebunnya" />
        <Property name="givenname" type="String" value="Mark" />
        <Property name="family_name" type="String" value="John" />
        <Property name="birthday" type="String" value="September 2" />
        <Property name="locality" type="String" value="Brussels" />
          <Property name="region" type="String" value="Region de Bruxelles-Capitale" />
          <Property name="country" type="String" value="Belgium" />
          <Property name="gender" type="String" value="male" />
      </Properties>
      <Annotations />
      <Target>
      </Target>
      <Mappings />
    </InstanceNode>
  </Nodes>
</ConceptualGraph>

```

Figure 5-5: World Specification Extract

5.3.1.2 Mapping Phase

The templates we need to have should be visualized. It is at the Mapping Phase, that a designer of the template will specify the objects and properties to be used to visualize these concepts that have been specified. We mentioned in Chapter 3 that one user may want a TV screen to view posts whereas another user may want wall painting for the posts, the other user may want to view photo frames of friends on the wall while another may want a box to have these photos in. This representation can be done using primitive 3D objects as we have seen in the case study in Chapter 4. Similarly with OntoWorld the designer will pick from an existing library of 3D objects to specify default objects that will be used for the virtualization

of each concept inside a 3D environment. Take for instance the Person concept in Figure 5-4 would be represented with a box. This is at the domain mapping. In the world mapping all instances of person would then be mapped to a box. Putting it differently, if we have five instances of person, then we would expect to see five boxes.

5.3.1.3 Generation Phase

At this stage, OntoWorld will generate the actual X3D code that we can use to visualize concepts in the Virtual World. If we follow the steps of VR-WISE using the OntoWorld tool, we can generate several templates to represent the concepts that we defined in our Person Communities Ontology. This is in addition to generating different Main World templates so that we can give one user an option to have a room as their Main World whereas another user may have the option to use a home compound as their Main World. These generated templates will then be wrapped under XSL and will make part of the templates repository seen in Figure 3-1.

5.3.2 Linking with VR-WISE

As discussed in the previous section, we can use VR-WISE OntoWorld tool to create templates to represent the different concepts in the Person Communities Ontology. Using the example in the section, the concept of Person had to be represented in a format that is interpretable by OntoWorld (See Figure 5-4). The implication of this is that we shall have to transform the Person Communities Ontology into the correct format to be interpreted by OntoWorld. This will start with creating a domain level ontology from which we can have a world specification. The rest of the phases namely mapping and generation of Virtual World will then be applied to successfully generate Virtual World template. Although the VR-WISE processes is clear and the OntoWorld tool enables us to generate these Virtual World templates basing on a Person Communities Ontology that conforms to the OntoWorld format, it does not sufficiently address complex object structures like how to represent a list of many friends. In the following section we look at means we can extend CoDePA which integrates with VR-WISE to handle complex static objects.

5.3.3 Extending CoDePA and VR-WISE

We earlier discovered that OntoWorld does a good job to generate a Virtual World if we know beforehand the kind and number of objects we are going to represent. As discussed

above, there is still a lack of proper representation and mapping for some forms of complex objects such as lists or grids. In Chapter 2 we stated that objects can be as simple as a brick, or as complicated as a YouTube video player or Facebook friend list. In our case study we have the Person Communities Ontology, in which there is a list of friends for example. It is possible that one person can have 5 friends whereas another person may have 500 friends. We thus would need a structured way to handle this kind of collection. OntoWorld can be improved using patterns to be able to generate better templates. The concept of patterns is seen in the CoDePA that provides a pattern-based framework to model Behavior for X3D worlds. We discussed earlier that CoDePA uses Generative Design Patterns (Pellens, et al., 2008) for specifying patterns of behavior that often occur in games. The Generative design patterns idea has been extended into what they call Visual Generative Design Pattern by expressing patterns using a graphical notation and by integrating them into existing graphical behavior (animation) modeling language.

The FOAF and SIOC 3D Approach case study templates were handcrafted and thus labor-intensive and missing support for reuse hence specifying more complex object may become more tedious. This can be avoided and hence designers' time saved with incorporation of Patterns that can be used in modeling complex objects like friends lists and more importantly if we follow the Visual Generative Design Patterns concept so designers can use an OntoWorld Visual Editor with these incorporated patterns. Currently the OntoWorld graphical editor tool that supports the VR-WISE approach lacks a way to represent these collections so that they can be reused by a designer.

In CoDePA behavior patterns are stored in a patterns library as seen in Figure 5-6. A Patterns User can specify behavior by instantiating patterns that are picked from the Behavior Patterns Library and customizing them to a particular context. More complex behavior can be constructed by combining different patterns. We proposed that CoDePA be extended with Static Patterns stored in the Static Patterns Library seen under the dotted border box. The Static Patterns in CoDePA will be defined by the Patterns Designer in a similar way the Behavior Patterns were defined.

We propose an extension to the OntoWorld Graphical Editor. This extension (See dotted boarder on Graphical Editor) is the Static Pattern Front-end with which a Pattern User shall

instantiate a pattern. As elaborated in CoDePA, this is done by selecting the appropriate pattern from the available library of patterns. When the static pattern is selected, the different graphical elements and connections that compose this pattern will be automatically dropped on the drawing canvas.

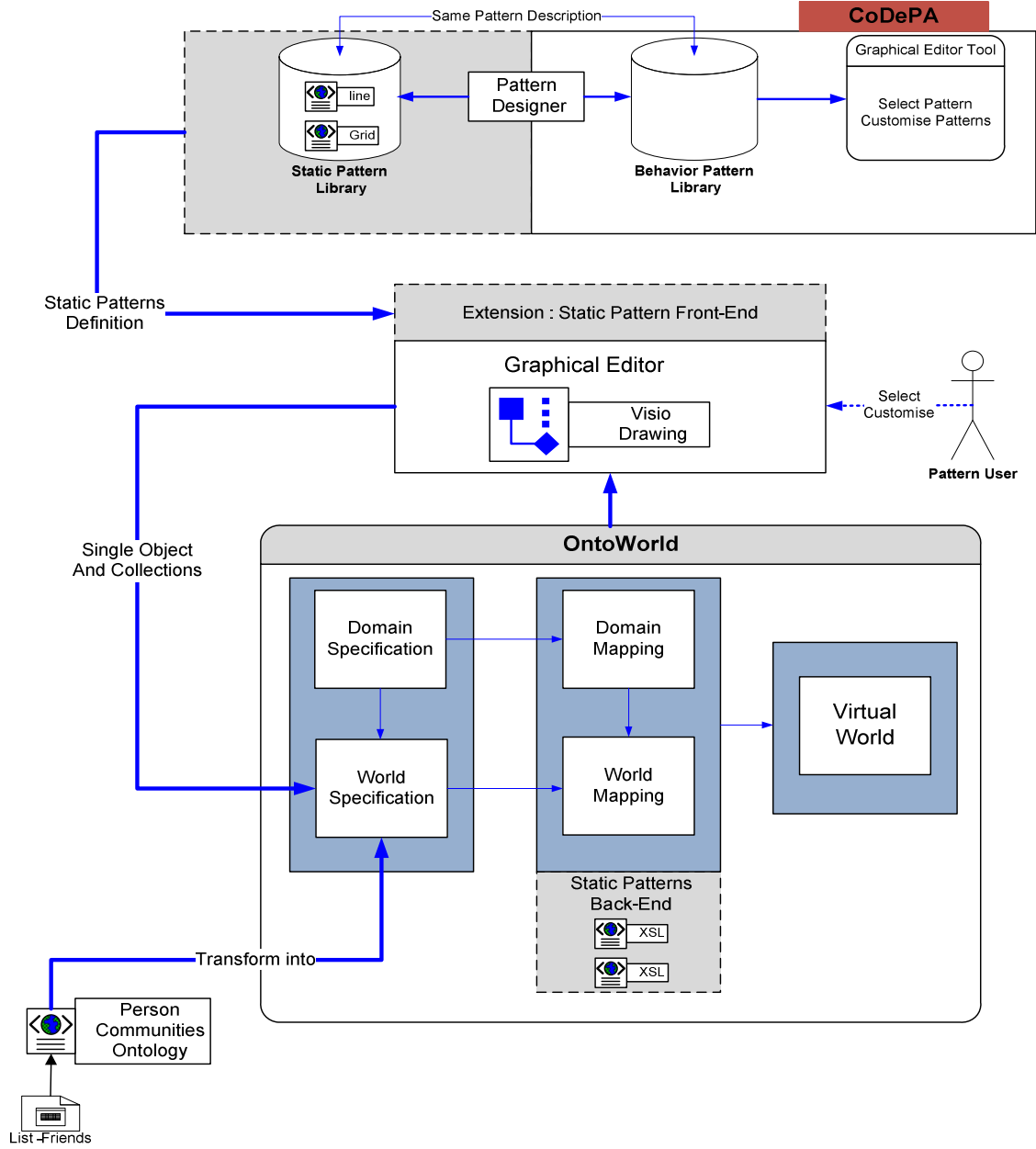


Figure 5-6 : OntoWorld Integration

Once the Pattern User has finished instantiation, the world specification will then receive a representation for both static objects and collections such as Grids. Having established that use of XSL can be used at the mapping stage, we propose a Static Patterns Back End

Extension to the mapping stage under OntoWorld. This extension with XSL will be used in generating Virtual Worlds with complex objects representation.

In order for FOAF and SIOC based 3D Approach Person Communities Ontology to be used with the improved OntoWorld tool, it will have to be expressed/converted into a format of the OntoWorld World Specification as discussed section 5.3.2.

6 Conclusion and Future Work

In the previous chapter we discussed how extensions can be made to the CoDePA and VR-WISE Approach in order to represent and make use of static patterns for complex objects. This chapter summarizes this thesis, looks at contributions and future work that includes insights into the implications of new social trends such as Facebook Open Graph.

6.1 Summary

In this thesis we developed the “FOAF and SIOC Based 3D Approach” that is used to transform content from different 2D online communities to a 3D Virtual World. To demonstrate the approach in a case study we developed an online tool to capture content from different online communities, store the captured content into an XML format that can potentially be a start of a big Mashup and build a 3D Virtual World. At the centre of the approach is the use of Semantic Web technologies such as the FOAF and SIOC ontologies which are used to represent social networks and social contributions respectively.

We introduced 3D online communities and the semantics for data portability in Chapter 1 and followed this with a detailed discussion of online communities, Semantic Web, ontologies, Virtual World, XSLT Transformation and related work such as SecondLife.com. The “FOAF and SIOC Based 3D Approach” was covered in detail in Chapter 3 followed by a case study in Chapter 4. Using the case study, we were able to observe the constraint of closed online communities’ that make data portability hard since they do not support open standards and semantic web approaches like FOAF and SIOC. Chapter 5 discussed the relation and integration of the “FOAF and SIOC Based 3D Approach” with existing approaches such as the VR-WISE approach using the OntoWorld tool to generate more visually appealing templates. In addition Chapter 5 elaborated on an extension of CoDePA with Static Patterns which OntoWorld can make use of through an extended front end graphical editor tool and back end XSL support for the mapping phase.

6.2 Contributions and Achievements

We developed the FOAF and SIOC Based 3D Approach to capture content from different online communities and transform this content to a 3D Virtual World. The approach makes use of Semantic Web FOAF and SIOC ontologies.

We also developed the necessary online tool support for this approach in order to build a 3D Virtual World case study using the captured data from different online communities. Using the case study we were able to prove that the use of this FOAF and SIOC based 3D Approach provides a solution to some of the existing data portability challenges among online communities.

We were able to transform the data from different online communities to a 3D Virtual World. This transformation and the entire approach was web based using XSL for the transformation. We thus prove that it is possible to have an online 3D Virtual World with rich content that is generated in real-time from different online communities.

The data retrieved from the different online communities was stored in a technology independent XML format with a view that this data can be a start of a big Mashup Application. Having been able to use this data for a 3D Virtual World, we believe that this data can be used for a big Mashup.

In order to ease the development of complex Virtual Worlds, we proposed extensions to CoDePA and VR-WISE OntoWorld tool. The extension on CoDePA is an addition of static patterns for complex static objects. The proposed extension to OntoWorld will use XSL as the back end at the mapping phase. In addition, a front end extension to the OntoWorld graphical editor is proposed to ease selection and customization of patterns.

6.3 Limitations and Future Work

With the case study, we have been able to realize that the “FOAF and SIOC Based 3D Approach” introduced in Chapter 3 can be used to get information from different online communities and transform this information to a 3D Virtual World. The tool support in the case study provided us with a means to realize the ability of the approach within the realms of a web browser and web server.

Though the approach elaborates how FOAF can be used to mine data from distributed social networks and to merge multiple online accounts held by a person, we observe that most of the current online communities such as Facebook, MySpace and Twitter do not yet have an open way to export FOAF or SIOC data at the time of developing the case study. This is still due to the walled garden model of online communities that has not yet adopted open

standards capable of expressing identity and other data implicitly. We can hope in the future that APIs such as that of Facebook will provide sufficient access to groups and communities of a user. With this in place we can retrieve and identify a user's data from various communities in addition to getting a comprehensive social network list of friends with the use of Semantic Web technologies like FOAF and SIOC.

The approach and case study focused on public accessible data thus we did not factor in how much data is available in the deep Web. For this to come to reality, there is a need for trusted identification. Hence in the future the use of OpenID or another identification mechanism will be investigated and added at some level of the approach and tool support. The identification will become more interesting to investigate in the future since we are currently seeing big social networks such as Facebook that has over 400M⁴⁵ users coming up with approaches such as Open Graph⁴⁶. The Open Graph innovation may still be at the micro stage thus we may not yet know its macro implication for the future. We cannot rule out that something we called "Facebook OpenID" in Chapter 2, may be seen in the future as the default identification mechanism deployed by other web services. Therefore, it will be informative to ascertain the micro and macro implication of this Facebook Open Graph.

At the stage of 3D transformation, the approach and case study uses 3D proto objects to demonstrate the concepts. In real deployment of the application tool, there is still a need of technical skills to develop richer X3D templates in terms on interactive functionality. Though this is the case, an extension of CoDePA with static patterns and an addition of a Visual Generative Design Patterns layer to the OntoWorld Graphical Editor proposed in Section 5.3.3 will provide better representation for collection of data.

In conclusion using the approach we can reduce on the time and technical skills required to generate content rich 3D Virtual Worlds in real time though the future of access to information of online communities will largely depend on the adoption of open standards such as FOAF and SIOC.

⁴⁵ <http://www.facebook.com/press/info.php?statistics>

⁴⁶ <http://developers.facebook.com/docs/opengraph>

Bibliography

1. Berners-Lee, T., Hendler, J. and Lassila, O. "The Semantic Web." *Scientific American Magazine*, 2001.
2. Biafore, B. *Visio 2007 Bible*. Wiley India Pvt. Ltd., 2007.
3. Bille, W., De Troyer, O., Kleinermann, F., Pellens, B. and Romero, R. "Using Ontologies To Build Virtual Worlds For The Web." *The International Conference on WWW*. Madrid (Spain), 2004: 683-690
4. Bishop, J. "Enhancing the understanding of genres of web-based communities: The role of the ecological cognition framework. *International Journal of Web-Based Communities*. *International Journal of Web Based Communities* (Inderscience Publishers) 5, no. 1 (2009): 4-17.
5. Brutzman, D. and Daly, L. *X3D: Extensible 3D Graphics for Web Authors*. 3rd Edition. San Francisco, California: Morgan Kaufmann Publishers, 2007.
6. Bryant, M. *Ignore Facebook Open Graph at your peril – this is Web 3.0*. April 21, 2010. <http://thenextweb.com/socialmedia/2010/04/21/ignore-facebook-open-graph-peril-web-30/> (accessed May 1, 2010).
7. Cook, A. *A case study of the manifestations and significance of social presence in a multi-user virtual environment*. Masters Thesis, Department of Curriculum Studies, University of Saskatchewan, 2009.
8. Daly, L. and Brutzman, D. "X3D: Extensible 3D Graphics Standard." *IEEE Signal Processing Magazine*, 2007.
9. Farquhar, A., Fikes, R. and Rice J. "The Ontolingua Server: A tool for collaborative ontology construction." *International Journal of Human-Computer Studies* (Academic Press, Inc) 46, no. 6 (1997): 707 - 727.
10. Fensel, D. *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. 2nd edition. Berlin, Germany: Springer, 2003.

11. Goble, C. A., and De Roure, D. C. "MyExperiment Ontology." *Proceedings of the 2nd workshop on Workflows in support of large-scale science (ACM)*, 2007: 1 - 2.
12. Golbec, J., and Rothstein, M. "Linking social networks on the web with FOAF: A semantic web case study." *Proceedings of the 23rd national conference on Artificial intelligence (AAAI Press)* 2 (2008): 1138-1143.
13. Gómez, P. *Knowledge Sharing and Reuse, In The handbook on Applied Expert Systems*. Florida USA: CRC Press, 1998.
14. Gruber, T. R. "Toward Principles for the Design of Ontologies Used for Knowledge Sharing." *ACM (Academic Press, Inc.)* 43, no. 5-6 (1995): 907 - 928.
15. Guarino, N. and Giaretta P. "Ontologies and knowledge bases: Towards a terminological Clarification." Edited by N. J. I. Mars. *Proceedings of 2nd International Conference on Building and Sharing Very Large-Scale Knowledge Bases*. Amsterdam, The Netherlands: IOS Press, 1995. 25-32.
16. Guillaume, E., Gandon F., and Buffa, M. "Semantic Social Network Analysis." *Proceedings of the WebSci'09: Society On-Line (Web Science Trust)*, 2009.
17. Harper, S. and Bechhofer, S. "SADie: Structural Semantics for Accessibility and Device Independence." *ACM Transactions on Computer-Human Interaction (ACM)* 14, no. 2 (2007).
18. Hendler, J., Shadbolt, N., Hall, W., Berners-Lee, T. and Weitzne, D. "Web Science: An Interdisciplinary Approach to Understanding the Web." *Communications of the ACM - Web science (ACM)* 51, no. 7 (2008): 60-69.
19. Jo Kim, A. *Community Building on the Web : Secret Strategies for Successful Online Communities*. 1st Edition. Berkeley, California: Peachpit Press, 2000.
20. Lunn Darren, R. *Sadie: Structural-Semantics For Accessibility And Device Independence*. PhD Report, Manchester, UK: University of Manchester, 2006.
21. Lynne, G. *OpenSocial Network Programming*. Hoboken, New Jersey: Wrox, 2009.

22. Muench, S. *Building Oracle XML Applications*. Edited by First. Sebastopol, California: O'Reilly Media, 2000.
23. O'Reilly, T. and Musser, J. *Web 2.0 :Principles and Best Practices*. O'Reilly Media Inc., 2006.
24. O'Connell, W. T. *Generative design patterns in heterogeneous virtual MIMD environments*. Illinois Institute of Technology, 1995.
25. O'Neill, N. *The Open Graph API: What Does It Mean?* 2009. <http://www.allfacebFook.com/2009/10/the-open-graph-api-what-does-it-mean/>.
26. Pellens, B. *A Conceptual Modelling Approach for Behaviour in Virtual Environments using a Graphical Notation and Generative Design Patterns*. PhD Dissertation, Brussels, Belgium: Vrije Universiteit Brussel, 2007.
27. Pellens, B., De Troyer, O. and Kleinermann, F. "CoDePA: A Conceptual Design Pattern Approach to model Behavior for X3D worlds." *Web3D '08 Proceedings of the 13th international symposium on 3D web technology (ACM)*, 2008: 91-99.
28. Pellens, B., De Troyer, O., Kleinermann, F. and Bille, W. "Conceptual Modeling of Object Behavior in a Virtual Environment." Edited by Eds. Fischer X. and A Bernard. *International Journal of Product Development (Inderscience Enterprises)* 4, no. 6 (2007): 626 - 645.
29. Pollock, J. T. *Semantic Web For Dummies*. Hoboken, New Jersey: Wiley. Publishing Inc, 2009.
30. Rehman, R. U. *Get Ready for Openid*. 1st edition. Hilliard, Ohio: Conformix Technologies Inc, 2008.
31. Rowe, M. and Ciravegna, F. "Getting to Me – Exporting Semantic Social Network Information from Facebook." *Proceedings of the ISWC 2008 Workshop Social Data on the Web*. Karlsruhe, Germany: University of Sheffield, UK, 2008.

32. Rymaszewski, M., Au Wagner, J., Wallace, M., Winters, C., Ondrejka, C, Batstone-Cunningham, B. and Rosedale, P. *Second Life: The Official Guide*. Alameda, California: Sybex, 2006.
33. Safko, L. and Brake, D. *The Social Media Bible: Tactics, Tools, and Strategies for Business Success*. New Jersey, USA: Wiley, 2009.
34. Segaran, T., Evans, C. and Taylor, J. *Programming the Semantic Web*. California, USA: O'Reilly Media, 2009.
35. Smart PR Corporation. "World First – Internet Enters The Third Dimension – ExitReality." (International Media Release) 2008.
36. Treadaway, C. and Smith, M. *Facebook Marketing: An Hour a Day*. Hoboken, New Jersey: Sybex, 2010.
37. Uschold, M. "Where are the Semantics in the Semantic Web?" *AI Magazine* (American Association for Artificial Intelligence) 24, no. 3 (2003): 25 - 36 .
38. Vince, J. *Introduction to Virtual Reality*. 1st edition. Springer-Verlag, 2004.
39. Vivaty. "Vivaty Create: Getting Started." 2007. <http://developer.vivaty.com/content.pdf> (accessed April 20, 2010).
40. W3C. "SIOC: Content Exchange and Semantic Interoperability Between Social Networks." September 2008. <http://www.w3.org/2008/09/msnws/papers/sioc.html#f2>.
41. W3C. *XSL Transformations (XSLT)*. Edited by James Clark. November 16, 1999. <http://www.w3.org/TR/xslt> (accessed Feb 25, 2010).
42. Web3D Consortium. "The Mission of the Web3D Consortium." *Web3D* . 2010. <http://www.web3d.org/about/mission/> (accessed Feb 12, 2010).
43. Yeliz, Y., Harper S., Goble, C. and Stevens R. "DANTE: Annotation and Transformation of Web Pages for Visually Impaired Users." *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters* (ACM), 2004: 490 - 491.

44. Zuckerberg, M. *Building the Social Web Together*. Facebook. April 21, 2010.
<http://blog.facebook.com/blog.php?post=383404517130> (accessed May 12, 2010).

Appendix A: PHP Snippet

```
<?php

// Create a new DOM Document object, this will create a new XML declaration for us.
// We can enter the version number and encoding (in that order), by passing
// two arguments to the object's constructor. By default, the XML
// declaration's version attribute will be set to "1.0"
$pdo = new DOMDocument();

// Creates an instance of the DOMImplementation class
$imp = new DOMImplementation();

// Creates a DOMDocumentType instance
$dtd = $imp->createDocumentType('graph', "", 'graph.dtd');

// Creates a DOMDocument instance
$dom = $imp->createDocument("", "", $dtd);

// Set other properties
$dom->encoding = 'UTF-8';
$dom->standalone = false;

// Create an empty element
// $element = $dom->createElement('graph');

// Append the element
// $dom->appendChild($element);

// Retrieve and print the document
$dom->saveXML();

// Here we create a new root element named foaf_sioc_combined.
$pRoot = $pdo->createElement('foaf_sioc_combined');

// We now add a new attribute to the foaf_sioc_combined element. We name this new
// attribute version and give it a value of foaf_basic.
$pRoot->setAttribute('version', 'foaf_basic');

// Finally we append the attribute to the XML tree using appendChild
$pdo->appendChild($pRoot);

// We repeat the same process again here, but this time we're creating
// the Person element.
$pPerson = $pdo->createElement('Person');

$pRoot->appendChild($pPerson);

// Create the main child nodes of person, these contain the information
// related to this xml file. Basically we're creating
// a new element for each node, the first argument specifies the name of
// the element we're creating, and the second specifies the text value
// of the node, for example, title would render as: <title>TalkPHP</title>
$pName = $pdo->createElement('name', $foaf_name);
$pGivenname = $pdo->createElement('givenname', $foaf_givenname);
$pFamily_name = $pdo->createElement('family_name', $foaf_givenname);
$pBirthday = $pdo->createElement('birthday', $foaf_birthday);
$pLocality = $pdo->createElement('locality', $foaf_locality);
$pRegion = $pdo->createElement('region', $foaf_region);
$pCountry = $pdo->createElement('country', $foaf_country);
$pGender = $pdo->createElement('gender', $foaf_gender);

// Here we simply append all the nodes we just created to the Person node
$pPerson->appendChild($pName);
$pPerson->appendChild($pGivenname);
$pPerson->appendChild($pFamily_name);
$pPerson->appendChild($pBirthday);
$pPerson->appendChild($pLocality);
$pPerson->appendChild($pRegion);
```

```

$person->appendChild($pCountry);
$person->appendChild($pGender);
//Here we retrieve array data for the list of online accounts that were found in the FOAF profile.

$accountServiceHomepage=$_POST['accountServiceHomepage'];
$accountProfilePage=$_POST['accountProfilePage'];
$accountName=$_POST['accountName'];

//Here we create the holdsAccount child node
$holdsAccount = $pDom->createElement('holdsAccount');

if(count($accountServiceHomepage)) {

    for($i=0;$i<count($accountServiceHomepage);$i++)
    {

        // here, we're creating an item element as our parent
        // and then creating and adding three child nodes to it.
        $pAccount = $pDom->createElement('OnlineAccount');
        $pServiceHomepage = $pDom->createElement('accountServiceHomepage', $accountServiceHomepage[$i]);
        $paccountProfilePage = $pDom->createElement('accountProfilePage', $accountProfilePage[$i]);
        $paccountName = $pDom->createElement('accountName', $accountName[$i]);

        $pAccount->appendChild($pServiceHomepage);
        $pAccount->appendChild($paccountProfilePage);
        $pAccount->appendChild($paccountName);

        // Here we simply append all the pAccount node we just created to the holdsAccount node
        $holdsAccount->appendChild($pAccount);
    }
}

// Here we simply append all the holdsAccount node we just created to the Person node
$person->appendChild($holdsAccount);

// Here we create a knows friends node.
$knowsFriends = $pDom->createElement('knows');

$friendsList=$_POST['friends'];

// Loop through each result from our friends list
if(count($friendsList)) {

    for($i=0;$i<count($friendsList);$i++)
    {

        // we're creating an item element as our parent child nodes to it.
        $pFriend = $pDom->createElement('Person');
        $pName = $pDom->createElement('name', $friendsList[$i]);

        // Here we simply append the Name node we just created to the Friend node
        $pFriend->appendChild($pName);

        // Here we simply append the Friend node we just created to the knows node
        $knowsFriends->appendChild($pFriend);
    }
}
$person->appendChild($knowsFriends);

// Set content type to XML, thus forcing the browser to render is as XML
header('Content-type: text/xml');

// Here we simply dump the XML tree to a string and output it to the browser

```

```
// We could use one of the other save methods to save the tree as a HTML string
// XML file or HTML file.

// $dom has entire XML, but, it's not clearly formatted, i.e there's no space or new lines in between tags, so I do this:
$pdo->formatOutput = true; // this adds spaces, new lines and makes the XML more readable format.

$xml_string = $pdo->saveXML();

//Here we write the file to a directory with the file name testFile.xml
$myFile = "person_xml.xml";
$fh = fopen($myFile, 'w') or die("can't open file");
fwrite($fh, $xml_string);
fclose($fh);

//Here we output the XML File to the screen
echo $xml_string

?>
```