Towards Formalizing Electronic Portfolios

Patrick Lougheed
School of Interactive Arts and Technology
Simon Fraser University
14th Floor, 13450 102 Ave.
Surrey, BC, Canada, V3T 5X3
patl@sfu.ca

Brittney Bogyo
School of Interactive Arts and Technology
Simon Fraser University
14th Floor, 13450 102 Ave.
Surrey, BC, Canada, V3T 5X3
bbogyo@sfu.ca

David Brokenshire
School of Interactive Arts and Technology
Simon Fraser University
14th Floor, 13450 102 Ave.
Surrey, BC, Canada, V3T 5X3
debroken@sfu.ca

ABSTRACT

Electronic portfolios have recently become a focus of serious research from many different quarters. However, there has been little attempt thus far to codify them in any logical method, nor has any direction been taken towards being able to reason about them. There are many possible uses for this, from aiding the user is creating the portfolio to helping employers sift through portfolio while searching for new employees. We present our research on the creation of an ontology for electronic portfolios, methods of instantiation the ontology and future planned experiments.

Categories and Subject Descriptors

I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods

General Terms

Portfolio Ontologies

Keywords

Electronic Portfolios, Ontologies, Reasoning

1. INTRODUCTION

1.1 What are Ontologies?

In computer science, ontologies are used to represent knowledge within a specified domain. They are typically comprised of types of objects, properties of objects and relationships between objects confined to that domain. By having a well created ontology, one can explicitly define or capture domain knowledge so that it has the potential to be computationally readable. Once an instance of the domain has been created, we can then extract information from the ontology and reason upon it.

"Ontological analysis clarifies the structure of knowledge. Given a domain, its ontology forms the heart of any system of knowledge representation for that domain. Without ontologies, or the conceptualizations that under-lie knowledge, there cannot be a vocabulary for representing knowledge. Thus, the first step in devising an effective knowledge-representation system, and vocabulary, is to perform an effective ontological analysis of the field, or domain" [4].

Ontology is a formal specification of knowledge in a domain. It formalizes conceptualizations [11, 6]. Ontology captures not only the commonalities among different conceptualizations in the domain but also formally establishes differences among those conceptualizations. In general, ontology is used as a mechanism to promote common conceptualizations. In this sense, we contend that one should focus on the process of capturing conceptualizations in the ontology rather than just the commonalities.

In a simplified sense, ontology provides an extendable and shareable framework to capture the common vocabulary in a domain. It includes machine-interpretable definitions of basic concepts in the domain and the relations that exist among them [8]. Presently, ontology is one of the popular knowledge representation techniques in AI.

Formally, ontology consists of entities, relationships, properties, instances, functions, constraints, rules, and other inference procedures. The power of ontologies rests with its ability to represent knowledge explicitly (as concepts, properties, and constraints); it's ability to encode semantics (as meta-data, rules, and other inference procedures); and it's ability to allow for a shared understanding of the represented formal knowledge within and in-between humans and the machines.

Recent surge in semantic web research has resulted in the evolution of a W3C standard - Ontology Web Language (OWL) . OWL enables the definition of domain ontologies, sharing of domain vocabularies, and the representation of the same at different levels of granularity. From a formal perspective, axioms and constructors in OWL capture the DL reasoning in terms of class consistency and consumption, in addition to other ontological reasoning.

1.2 What are ePortfolios?

"An ePortfolio is a personal digital collection of information describing and illustrating a persons learning, career, experience and achievements. ePortfolios are privately owned and the owner has complete control over who has access to what and when. ePortfolios contents and services can be shared with others in order to support Prior Learning Accreditation and Recognition (PLAR), complete or replace exams, reflect on ones learning or career, support continuing professional development, plan learning or search a job" [5]. Furthermore, ePortfolio development and content can greatly impact goal creation and achievement as well as facilitate self-regulated learning activities.

While the concept of portfolios is not a new one, a recent expansion of their scope from merely a paper-based job searching tool to a digital educational tool has greatly enhanced both the breadth and depth of the research on the subject. A number of new tools have been created, both as research platforms and for commercial use, helping users create their own portfolios. These systems are typically implemented as editors around a set of artifacts that the user has created, and their responses to them. These artifacts and responses can then be placed into a coherent structure and ultimately placed somewhere where others can view the portfolio and give feedback upon it.

Portfolio implementation varies greatly differing from individual selective showcase portfolios to institutional reflective portfolios. Individual creative portfolios can be used to record accomplishments, set goals or simply to provide a complete digital reflection of ones actions. Institutions can use ePortfolios to create a picture of all they have to offer with examples provided by their members. Traditionally portfolios have been used in the arts as part of marketing artists work to potential employers or clients. ePortfolios have recently become more popular in the wider community as learning tools, knowledge retention mechanisms and alternative forms of assessment.

1.3 Why capture ePortfolios?

ePortfolios are a rich and plentiful source of data about individuals and organizations. For example, with successful parsing of the information within an ePortfolio an intelligent tutoring system or a user model could be better informed about a students actions and performance. This relationship can also go the other way, where students can use information garnered by their user model within their portfolio as dynamically updated information or simply as a point of reference.

Unfortunately, one current problem with the state of the art in electronic portfolio is that of user buy-in. While many institutions and corporations have adopted the idea of electronic portfolios whole-heartedly from a management perspective, there are still a plethora of issues keeping users from using portfolios to their advantage. These reasons include a lack of communication from those providing the service with regards to the benefits of having an electronic portfolio, and a lack of the critical mass of users and readers of portfolios. Building and maintaining portfolios can prove to be time consuming particularly if they are not structured well. Furthermore, garnering meaningful data out of a portfolio also takes a significant time-commitment.

In addition, the field of educational research surrounding the benefits and pitfalls of ePortfolios is in its infancy and thus the uses and structures are evolving. There are several different types of portfolios created for many different purposes, abiding by neither standard nor any common definition of what makes a portfolio.

It is our opinion that proper design and implementation of an ePortfolio creation and maintenance tool can solve some of these problems. For example, ePortfolios created in educational environments should not have to be subject to the cold start problem, nor should they be particularly complex to maintain as, in some cases, their structure and content can be obtained through alternate means. Furthermore ePortfolios currently exist to aid the user, an ePortfolio tool needs to encourage this personal motivation and customization while still making the ePortfolio abide by some standard (that will ultimately arise from the chaos).

The benefits of having access to such a rich knowledge base is key in improving delivery and customization of educational technology. Our aim is to build an ePortfolio tool that meets these goals and overcome some of these issues.

2. BACKGROUND

Our tool, SPARC (the Student Portfolio Architecture and Research Community), is currently in it's second major revision, and is being both in production use at Simon Fraser University for more than 400 students, and as a platform for research in various different areas. SPARC is implemented under the principle of 'everything is an artifact' - files, URLs, reflections on these, other people's feedback, etc. It can manage multiple portfolios for a single user, and has an artifact repository. The back-end is based on SOAP and web services, allowing an infinite number of front-ends, of which we have implemented two, one web-based and one a local Java program.

Portfolio creation in SPARC (and in most other systems) generally happens cyclically, in about three different phases. The first phase is artifact cataloguing, where the user (and possibly the system) adds artifacts to the portfolio tool. The second is laying the portfolio out - placing the artifacts in a tree structure, reflecting on the artifacts, etc. - while the final phase is publishing the portfolio. The first two phases happen simultaneously in many cases, and the entire process repeats itself based on external feedback, new artifact creation as time passes, or different requirements of the portfolio. Also note that a user can have multiple portfolios simultaneously at different points in the process.

We previously attempted an upper-level ontology for electronic portfolios [2]. We have substantially updated and modified that during our research; it was felt that the previous version was inherently over-restricted as well as too specific to the original implementation of the SPARC tool. Large portions have also been added that were not present beforehand, including the capability to capture author information and system interactions.

3. COMPONENTS OF THE EPORTFOLIO TAXONOMY

At the most simplistic level, all portfolios are collections of artifacts that may or may not be related to other artifacts. These artifacts can be broken down into a number of standard types that better reflect their purpose and content. Breaking them down also allows us to infer more from them than we would be by looking at them solely as artifacts. Artifacts can also have multiple different authors,

Because portfolios, at least assessment portfolios, are considered forms of alternative assessments, any ontology created must take into account alternative assessment standards. Tanimoto proposes a taxonomy of assessment items so that information pertaining to alternative assessment can be classified [12]. Portions of this taxonomy relate closely to portfolios, but are not sufficiently deep to allow useful classification of many types of artifacts.

The simplest of portfolios contain examples of projects created by the author and no other information. However, more complex portfolios can contain, in addition to these examples, written reflections about how the project turned out, explanations of why the project was created (i.e. for an assignment), and even feedback from peers. Each of these elements can be defined as a type of artifact.

The first step in creating the ontology is creating the initial taxonomy, which is comprised of the elements discussed below. The final hierarchy is given in Figure 1.

3.1 Portfolio Types

There are a large number of different portfolio types, broken down into four sub-categories - *dossier* portfolios, *training* portfolios, *reflective* portfolios and *personal development* portfolios [10]. The categories are based on two main criteria. The first is whether or not the portfolio creation is mandatory, voluntary, or somewhere in between (for example, not mandated, but highly suggested). The second represents the selection process use during the creation of the portfolio, with a scale from specific selection portfolios to learning portfolios encompassing a full range of work. The essential difference here is that selective portfolios only show off the best that the author has created, while a learning portfolio will include the entire range of the author's work, showing the learning process and documenting what they took out of that process. Using these two criteria as axes, we can visualize the gamut of portfolio types and the categories they fall into (Figure 2).

Dossier portfolios include those types that are mandated and selective, often having a set outline of the artifacts that should be included. Examples of uses of dossier portfolios are those used in the medical [7] and graphics design businesses, where the potential employer or client desires a well laid out view of the author's best work, usually with certain required elements.

Training portfolios are mandated portfolios that document learning attempts that the author has undertaken for a particular curriculum. For example, it may be necessary for a course that the author is taking that they document their work for the course and reflect on what they've accomplished.

Reflective portfolios are strictly voluntary, while also being fairly highly selective. Authors choose their best work, tracking progress and improvement over time with evidence and allowing for reflection and self-appraisal. Selection occurs with the authors choosing those portions which best define their improvement, potentially showcasing the starting point, successful improvement attempts, and what they learned, and how this applies in different scenarios and environments.

Personal development portfolios are a more thorough collection of work over time, without the selectivity that comes with reflection portfolios. They are long-term portfolios, often focusing on internal development of the individual rather than their external development. These portfolios can serve as as the launching point for dialogue regarding the author, their path and future self-improvement.

Figure 1: An Ontology of Electronic Portfolios.

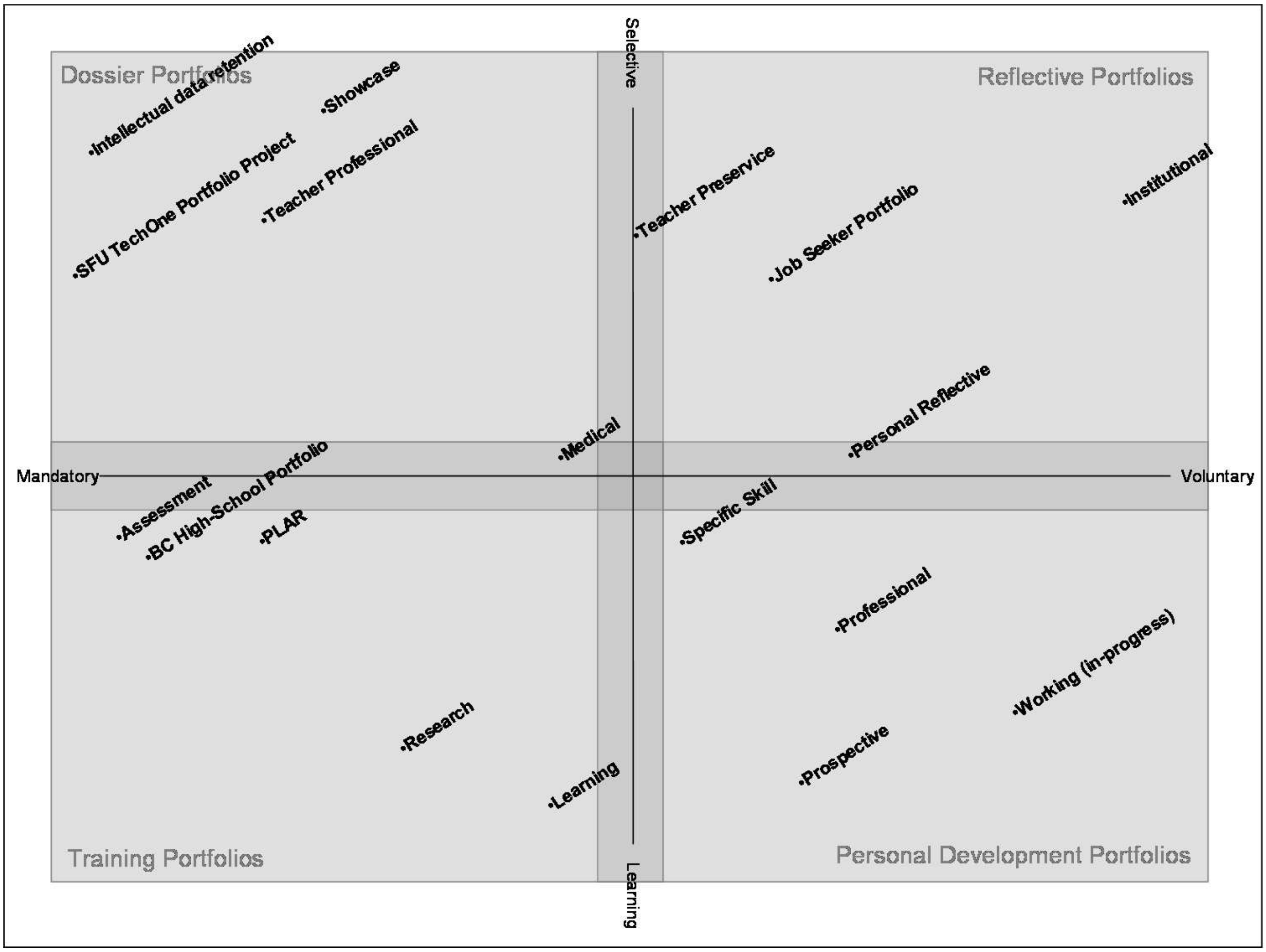

Figure 2: Portfolio Types.

Within these categories we have identified a further 18 specific types of portfolios - the actual number of types is, of course, limitless, varying within the dimensions given above. There are a few types that are very specific to our experience - the "SFU TechOne Portfolio Project," a two-semester project for a first-year cohort program at Simon Fraser University's Surrey, British Columbia campus, as well as the "BC High School Portfolio," based on the graduation requirements of the Ministry of Education in British Columbia [1] - while others are more general types.

3.2 Portfolio Process

In order to create a portfolio certain processes must be undertaken. Each type of portfolio will contain a different subset of these for the overall portfolio creation process. Each process represents a different method of going about the system creating the portfolio, from collecting the initial artifacts into the portfolio to organizing the content to laying out the published version of the portfolio visually.

3.3 Author Information

A portfolio has at least one author, and can have a number of different authors. Each author, conversely, can have many different portfolios and artifacts, or none. For each author we can store some basic information, with regards to their educational and professional history and other information that may make use of.

Authors are further split into two categories - learners and mentors. The differentiating factor is that learners author portfolios (and own at least one) while mentors author external evaluations. An individual can qualify as both a learner and a mentor, as well.

3.4 Artifacts

A portfolio is inherently a collection of artifacts with some relationship to the author. The artifacts can vary, as can their relationships, but these can be generalized by splitting artifacts into two camps - evidence artifacts, and description artifacts. Description artifacts can be further broken down, yielding different types of decriptions as well as evaluations, both internal and external.

As well, artifacts can have relationships to one another; descriptions provide context to evidence or to other descriptions, while evaluation artifacts provide feedback on other artifacts. These relationships can be generalized as well, and we will expand upon that shortly.

3.4.1 Evidence Artifacts

Evidentiary artifacts show the portfolio's audience the record of what has been achieved. To put this in perspective, the difference between a portfolio and a traditional resumé is that a resumé tells the audience about what has been attained; the portfolio will show the audience the proof. With this in mind, the evidence can be a wide variety of things - an image, a document, a sound file, a URL to a created website, or even a pointer off-line.

3.4.2 Description Artifacts

Description artifacts provide the context that gives meaning to the evidence. Evidence without context is meaningless, so descriptions play a fundamental role in the portfolio.

We have subdivided description artifacts into *activity* descriptions, *criteria* descriptions, *general* descriptions, and *evaluations*.

Activity descriptions provide information about activities performed, the environment they were performed in, and what the outcome was. For example, they could be a description of a physical activity undertaken for a class, or the outline of an experiment run to test a hypothesis.

Criteria descriptions detail the criteria used in judging another artifact. This could be anything from the evaluation method of a classroom instructor, to the null and testing hypotheses for the aforementioned experiment, to describing how a student is critiquing their own work.

General descriptions are "unclassified" - if not specifically an activity or criteria description, they are treated as a general description.

3.4.3 Evaluation Artifacts

Evaluation artifacts provide some feedback on other artifacts. We split them into two categories - external evaluations (coming from someone aside from the authors of the artifact) and internal evaluations (coming from one or more of the authors).

Note that evaluation artifacts can evaluate other evaluations, as well. For example, consider the case of a set of description and evidence artifacts created as part of a creative writing assignment for a university course. The course teaching assistant may initially evaluate the assignment, and the course instructor may later evaluate the TA's evaluation.

3.4.4 External Evaluation Artifacts

External evaluation items can take many forms. The typical grade school term-ending report card is at the simplistic end of the scale, while complex evaluations such as the revisions to a thesis or a detailed breakdown of the problems with a submitted paper are good examples of external evaluations. These evaluations can come from any number of individuals - friends, family members, teachers, supervisors, employers, even anonymous reviewers.

3.4.5 Internal Evaluation/Reflection Artifacts

Internal evalution is similarly multi-modal. In many cases it can take the same form as an external evaluation, especially when following a set of pre-defined rubrics for the evaluation. However, there is a one possible form that is fundamentally different - reflection.

Reflection can be viewed as critical self-appraisal; it is a detailed and thorough look at the process one took and the outcome one received to see why that path was chosen, how the path turned out, how it could be improved, what was learned, and what pitfalls, problems or wrong turns were made or encountered.

4. TURNING THE TAXONOMY INTO AN ONTOLOGY

Once we have the taxonomy created, we can begin to flesh out the ontology by adding relationships between the different objects, and giving objects properties and rules.

We are using the Protégé ontology editor, with OWL as the backend language. Racer is used as the inference engine.

4.1 Relationships

There are a number of different relationships that exist within the ontology. Aside from the "is-a" relationships that can be seen directly from the taxonomy, several of these relationships are fundamental to portfolios.

For the learner object, there exists a "hasPortfolio" relationship to one or more portfolios. Conversely, mentors have "provideExternalEvaluation" relationships to external evaluation artifacts.

Portfolio objects consist of three basic relationships - "hasArtifacts" is a collection of all the artifacts present in that portfolio, "hasCreationProcess" outlines the processes used to create the portfolio, and "hasPortfolioOwners" points to the author or authors of the portfolio, and is the inverse of the learner "hasPortfolio" relationship.

Processes have the "constructsPortfolio" relationship, the inverse of the portfolio's "hasCreationProcess." One special subclass of Process is PortfolioCreationProcess, a container object for other processes, collecting them into a single higher-order object. This allows for restrictions for portfolio types; see the Example section below.

Artifacts at the top level have two relationships - "belongToPortfolio" shows us the portfolios the artifact is a part of, and is the inverse of "hasArtifacts" from portfolio; and "hasArtifactAuthors" points to the author(s) of the artifact.

Descriptive artifacts have several other relationships, as well. They have a "canBeUsedAs" relationship to evidence artifacts, as they can often be used as direct evidence rather than just as context. As well, they can "referToArtifact" to provide information or context about several other artifacts.

Evaluation artifacts build on top of these relationships again. They can "referToPortfolio," and give evaluation on an entire portfolio rather than just an artifact. External evaluation artifacts also have a "hasEvaluationAuthor" relationship detailed who left the evaluation, and self-evaluation topics have a "hasSelfEvaluationAuthor" relationship that accomplishes the same task (the difference being a rule in force on self-evaluation artifacts; see below).

4.2 Properties

Each superclass of objects has it's own set of properties in addition to relationship.

Artifacts, for example, all have a "Title," and descriptive artifacts have a "WrittenDescription." Evidence artifacts have ae "EvidenceType" property, which is an enumerated type consisting of image, video, audio, document, multimedia file, or URL, as well as a "LocationURI" pointing to where the evidence exists. Portfolios also have a "Title" property.

Authors have a number of different properties. These include "Age", "ContactInformation", "EducationalHistory", "EducationalStatus", "Name", "ProfessionalHistory", "ProfessionalStatus", "Sex", and "UserID."

Processes have no properties.

4.3 Rules

Artifacts have very few rules, with one notable exception. For self-evaluation artifacts, the author(s) of the self-evaluation must be a subset of the authors of the artifact being evaluated.

For authors, they only qualify as learners if they have at least one portfolio.

Processes and portfolios in general have no rules, but PortfolioCreationProcesses and lower-order portfolio types may have rules. We have a few examples in our ontology, specifically tying certain creation processes to certain portfolio types. See below for one example of this.

4.4 Example

As an example of our ruleset in the ontology, consider the case of a specific portfolio type called the "SFUTechOne." This type has an associated process with it, called the "SFUTechOneProcess."

The "SFUTechOnePortfolio" has a restriction that it must have exactly one author. As well, it has the restriction that all of the creation processes that are used to build the portfolio must be of the type "SFUTechOneProcess."

The "SFUTechOneProcess," on the other hand, has a rule that states that all the portfolios with which it has a "constructsPortfolio" relationship must be of type "SFUTechOne." The process may be used to create only those portfolios, and those portfolios can only be created using this process. From this, we can infer that if someone is creating a "SFUTechOne" portfolio, that they are using the corresponding process, and suitably constrain the creation of the portfolio.

The process is a higher-order process encompassing other processes, specifically the "Categorization," "Collection," "Publish," "Reflection," "Revision," "Selection," and "VisualOrganization" processes. Effectively, this means that an SFU TechOne portfolio does not involve the identification of marketable areas, identification of skill areas, logical organization or maintenance processes.

5. INFORMING THE ONTOLOGY

Once we have an ontology created, the next step is somehow getting data that can be inputted into the ontology, and without this the ontology is merely a theoretical exercise. The input needs to be relevant to the ontology - there is no point telling the ontology about things it doesnt know about or things it cannot reason upon - but also as unobtrusive to the user as possible. We cannot ask the user to spend half of their portfolio creation time putting things in the proper format for our ontology and continuing to require information of them; in order to make the ontology truly useful, we have to find a way to gather as much of this information as we can ourselves.

There are a number of different ways we can instantiate the ontology, and the information we use will need to come from a variety of different sources. These include manual, semi-automatic and fully automatic instantiation, and the data can either come from the SPARC system, an external system, or from the user - noting of course that the user, as stated, will be a minimal source of information and they will not be conscious of their contribution.

5.1 Instantiating the Ontology

5.1.1 Manual Instantiation

The ontology can be manually instantiated via entries of property values into an ontology editor - for example, Protégé. This is useful only for testing portions of the ontology or attempting to create useful inferencing rules, and is not viable over the long term. Manual instantiation requires us to have access to the necessary information about the portfolio creator - about their artifacts, their interactions with the system, and their intentions when creating the portfolio.

There are several limiting factors to manually instantiating the portfolio ontology. The first is that the user is likely to have thousands upon thousands of individual interactions with the system, from clicking buttons to selecting files for uploading to editing content. This makes it prohibitively expensive with regards to time. The second is that this would have to be created manually for each individual user; in any multi-user system with tens or hundreds (or more) of users logged on and interacting at any given moment, it quickly becomes impossible to keep up with the incoming data. One user must be singled out for testing purposes.

Manual instantiation is by far the most simplistic model to implement, as the portfolio software needs to have no knowledge whatsoever of the ontology or how to push information to it. Some type of logging framework is required, so that the person ultimately inputting the information into the ontology can gain access to the necessary data. The simplest example is a messaging framework that dumps a message to a log file for each user action, with various properties for the message. For example, when the user presses a button, we can generate a "button pushed" message, with an indication as to which button was pressed, how it was triggered (say, mouse or keyboard shortcut), and what the outcome of the interaction was (an error message was thrown, a dialog was created, etc.).

5.1.2 Semi-Automatic Instantiation

Semi-automatic instantiation is used for the initial discovery and creation of rule sets to be later used for inferencing. It combines the immediate recording of data from automatic instantiation with the ability to have a closer look at the data while inputting it. For example, if we know that a certain individual is working on a specific portfolio type, we can record their interactions and then make slight alterations to it in an attempt to get a rule to fire.

This type of instantiation again requires use of an ontology editor, but here we must have the portfolio system itself inputting information into the ontology and saving it, so that we can later manually edit. The messaging framework we created in the first place can be modified to generate inputs into the ontology rather than dumping information to the log file, skipping the human interaction that was necessary beforehand. This method also allows for gradual integration of the ontology input into the system, rather than forcing the programmer to attempt to convert from logging to ontology input all at once.

5.1.3 Automatic Instantiation

Automatic instantiation of the ontology is, of course, the ultimate goal. This allows for the portfolio system to input all the information by itself, on the fly, rather than relying on human intervention to tell the ontology about the data. By this point most of the rules and constraints will be laid down into the ontology itself. One major advantage of this method is it allows for immediate updating of the ontological information about a user; we can input new data from their interactions and see if any of our inferences about them have changed, and possibly alter the method, frequency or content of our interactions with them.

Getting to the point of being able to instantiate automatically means that we must be able to incorporate the information coming into us from other sources, even if those sources aren't necessarily ontologically aware. For example, one of SPARC's features is the ability to recieve artifacts from sources other than the portfolio creator; these sources could include assignment submission systems, learning management systems, other users (for group artifacts) or even student information systems. To ask all of these systems to become fully aware of the ontology may be a laudable goal, but not one likely to be reached any time soon. In the meantime we must be able to take the data and metadata they push to us and perform the transform and input of that information ourselves.

5.2 Gathering Data

5.2.1 SPARC Internals

SPARC itself will generate the vast majority of the information that will be placed into the ontology. This includes the interactions data, information about some of the artifacts that are uploaded (those that are uploaded or modified via the SPARC interface) and information on the user of the system.

The interactions data consists of the different steps the user follows in creating or editing their portfolio. These steps include pressing buttons, modifying character content, checking boxes, etc. Each of the different interfaces for SPARC will have a slightly different variation upon the central theme of interactions, but the vast majority of the information will be applicable across these different front-ends. This information also allows us to run experiments to see what effect the interface can have on the portfolio experience.

5.2.2 External Systems

External systems account largely for the information regarding artifacts that can be pushed into the ontology, as their primary responsibility is pushing artifacts to SPARC without user intervention. Again, we cannot assume that these systems know anything about the portfolio ontology, so any information that gets inputted will actually come from SPARC. We can infer some basic information about the user's actions from what the external system provides us, and we place that into the ontology.

This information generally includes information on what the user uploaded and via what method. Some extra information includes what other individuals can have access to the artifact (if the work was generated by a team, for example) and why it was uploaded, such as a class designator. Other systems, such as student information systems that could be tapped into, can provide information along more biographical lines, such as past learning experiences that can give us further context that could help the ontology make better inferences.

5.2.3 User

The user can, unwittingly, be the source of some information for the ontology, if we tell the program where to look. There are two specific places we can look - the names and metadata of the files that the user uploads, and the relationships that can be inferred from the user's interactions with artifacts.

For example, if the user uploads a file with a specific filename - say, "CS 101 - Assignment 1.doc" - and places that artifact such that it develops certain relationships within the ontology, and later uploads another file named "CS 101 - Assignment 2.doc"; we can then infer that perhaps we should create relationships for this new artifact, as well. The more similarly named or placed artifacts the user uploads, for example, the better we can become in helping the author by creating some relationships for them.

6. FUTURE WORK

There are a number of plans for future work utilizing the ontology we have layed out. First, we plan to verify that the ontology is completely usable by integrating it with the SPARC ePortfolio system. Once this is complete, we can expand the reach of the ontology by adding connections to the external data sources. Specifically, we are targetting the WebCT learner management system and a home-grown Course Management System, both in use at Simon Fraser University's Surrey campus.

As well, we wish to allow for knowledge transfer with other systems, by connecting to the Interoperable User Model framework [3]. Using the ontology and data we can gather from connecting to other systems, we wish to build into SPARC functionality to allow for mixed-initiative interactions with the portfolio authors.

Finally, we wish to create a learning object repository from the ontology utilizing the eduSource Communication Language [9] to connect to other similar repositories and allow for searching of portfolios.

7. CONCLUSIONS

We have presented our initial revision of an ontology for electronic portfolios. We have fleshed out a taxonomy, as well as relationships between different elements, established rules for constraining objects and inferring information about others, and initial properties for objects. We believe this to be a crucial first step to making electronic portfolios more usable and useful to potential users.

In the larger picture, the creation of an ontology for electronic portfolios will allow for an expansion and transfer of knowledge outwards from this area. It will allow for other systems to glean information from electronic portfolio systems about users and portfolios, a huge step given the amount of information that a portfolio can contain or infer about a user.

On the semantic web, for example, having a portfolio ontology could allow for employers to formulate a query to find all individuals that meet certain criteria from those that applied for a job based on their portfolios; potential employees could do the same thing in reverse, finding any jobs that match their skills.

Assuming that ontologies promote the use and the extension of a common formal conceptualization in each domain one may assume that simply employing ontologies in web-based systems would realize the goals of Semantic Web. Unfortunately, the world of Semantic Web is much more complicated than to be solved by such a simplistic notion. As we mentioned earlier the centrality of ontology is in the process of capturing conceptualizations in the ontology. In a community of users interacting within a semantic web application that revolves around a common ontology, it is inevitable that inconsistencies arise in the ontology among multiple users over a period of time. Maintaining such inconsistencies in the ontology is quite intractable and remains the foremost challenge in Semantic Web.

8. ACKNOWLEDGMENTS

The authors acknowledge the support of the National Sciences and Engineering Research Council of Canada via the Learning Object Repositories Network grant and the Social Sciences and Humanities Research Council of Canada via the Learning Kit Initiative on the New Economy grant.

The authors also wish to thank Mayo Jordanov, Marie-Claude Lavoie and Jessica Ramsay for their efforts in the development of the SPARC tool, and Liam Doherty and Jurika Shakya for comments on the initial drafts.

9. ADDITIONAL AUTHORS

Vive Kumar (School of Interactive Arts and Technology, Simon Fraser University, email: vive@sfu.ca).

10. REFERENCES

[1] British Columbia Ministry of Education. Graduation portfolio assessment and focus areas: A program guide. http://www.bced.gov.bc.ca/graduation/portfolio/moe_grad_portfolio_p1_p2%.pdf, 2004.

[2] D. Brokenshire, B. Bogyo, and V. Kumar. Towards an upper level ontology for eportfolios. In *Proceedings of ePortfolio 2004*. European Institute for E-Learning, 2004.

[3] D. Brokenshire, J. Shakya, and V. Kumar. Providing information for mixed initiative interaction via interoperable user modelling. Accepted at the *AAAI Fall 2005 Symposium* track on *Mixed-Initiative Problem Solving Assistants.*, 2005.

[4] B. Chandrasekaran, J. R. Josephson, and V. R. Benjamins. What are ontologies, and why do we need them? *IEEE Intelligent Systems*, 14(1):20–26, 1999.

[5] European Institute for E-Learning. Europortfolio. http://www.eife-l.org/portfolio.

[6] T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[7] J. McKimm. Using portfolios in medical education. Imperial College of Medicine, 2001.

[8] N. F. Noy and D. L. McGuinness. Ontology development 101: A guide to creating your first ontology. Technical Report KSL-01-05, Stanford Knowledge Systems Laboratory, Stanford University, 2001.

[9] G. Richards and M. Hatala. Linking learning object repositories. *International Journal of Learning and Technology*, 1(4):398–409, 2005.

[10] K. Smith and H. Tillema. Clarifying different types of portfolio use. *Assessment & Evaluation in Higher Education*, 28(6):625–648, 2003.

[11] J. F. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations.* Brooks Cole Publishing Co., Pacific Grove, CA, 2000.

[12] S. L. Tanimoto. Towards an ontology for alternative assessment in education. Technical Report TR-98-09-04, Dept. of Computing Science and Engineering, University of Washington, 1998.