

INTEROPERABILITY

AND THE EXCHANGE OF HUMANLY USABLE DIGITAL CONTENT ACROSS A GLOBAL ECONOMY

Richard A. Vines

Project Lessons – Strategic Solutions,
Melbourne, Australia

Joseph M. Firestone

Knowledge Management Consortium International
Executive Information Systems, Inc.
Alexandria, VA, United States

We are very grateful to David Vines and Denis Noble (Oxford University), Paul David (Oxford and Stanford Universities), David Cleevely (Chairman: CFRS – UK) and Keith van Rijsbergen (University of Glasgow) for their helpful comments on early drafts of this paper.

Addresses:

Richard Vines: Project Lessons – Strategic Solutions; Surrey Hills, Melbourne, Australia;
Email: plessons@netspace.net.au;

Joseph Firestone – Executive Information Systems; Alexandria Virginia, United States:
Email: eisai@comcast.net;

PREFACE

This paper has been written with two very specific ends in mind.

The first is simple. We aim to make a small contribution to a culture of spirited problem solving in relation to the design and management of complex and open information networks that form part of the vitality and sustainability of any modern knowledge economy. For example, there may be an interest in the information networks associated with government and government funded partnership collaborations in areas encompassing integrated health, well being and community services. Or there may be an interest in other types of complex service networks encompassing insurance brokering or the design, manufacture, retail and after sales service. Or the interest may encompass the monitoring and management of compliance information in response to the implementation of regulatory standards across any sort of domain of activity.

The second end is more complex and relates to the possibility that any debate about technical and social aspects of interoperability and content exchanges is not restricted to technical and IT related personnel. The use of mass digital technologies and the widespread adoption of the internet have been with us for less than twenty years. As the pace of adopting these new systems increases, it remains the case that remarkably little has been written about emergent options becoming available in terms of how we envision, negotiate and implement new information management systems across multiple sectors that make up all knowledge economies. This paper aims, in a small way, to address this current situation by tackling one particular technical concern – namely the emerging explosion in the use of eXtensible Markup Language (XML). The rise of XML is seen as an important solution to the challenges of automated (and semi-automated) information and data processing. However the problem of interoperability between XML schemas remains a global challenge that has yet to be resolved.

Both of us have a background in, and understanding of, the theory and practice of knowledge management (KM). In highlighting this – we should not be misunderstood as to the focus of this paper. We are not writing about KM itself. Rather, we are concerned about the nature of support systems that provide a foundation for effective KM. But in tackling this challenge, we think that the principles of KM themselves can be used in the evaluation of particular theories of interoperability as discussed herein. We trust that our aspirations are moderately successful to this end.

Richard Vines and Joe Firestone

TABLE OF CONTENTS

PREFACE	2
EXECUTIVE SUMMARY	4
1 INTRODUCTION	5
2 THE TRANSFORMATION OF DIGITAL CONTENT	8
2.1 A current approach to transformation and interoperability.....	8
2.2 The structure of the automated translation problem	9
2.3 Addressing the problem of incommensurability.....	11
2.4 System components of the translation / transformation architecture.....	14
2.5 Conclusion: An interlanguage as a theory of translation	16
3 THE XML-BASED INTERLANGUAGE APPROACH: TWO EXAMPLES	17
3.1 The XML-based interlanguage approach.....	17
3.2 Introduction to the two examples of XML-based interlanguages.....	17
3.3 The CGML translation/transformation architecture.....	18
3.4 How the CGML system works.....	20
3.5 The COAX translation / transformation architecture.....	24
3.6 How the COAX Transformation System Works	25
4 THE ONTOLOGY-BASED INTERLANGUAGE APPROACH: ONTOMERGE	27
4.1 The Ontology-based interlanguage approach using OWL.....	27
4.2 OntoMerge: An Example of an Ontology-based Translation System.....	28
4.3 OntoMerge and the translation of XML content.....	30
4.4 How the OntoMerge system works.....	33
4.5 Differences in approach	35
5 CHOOSING AMONG SYSTEMS	36
5.1 The Commensurability Creation Load.....	37
5.2 System design to integrate the use of human interpretative intelligence.....	39
5.3 Facilities for the use of human interpretative intelligence	40
6 ADDRESSING THE TRANSLATION PROBLEM: EMERGENT POSSIBILITIES	50
6.1 Possibility One. No choice - A patch and mend architecture prevails	50
6.2 Possibility Two. Choice to maximize the automation of translations.....	50
6.3 Possibility three. Choice to embed human interpretative intelligence.....	54
7 CONCLUSION	55
8. REFERENCES	60

EXECUTIVE SUMMARY

This paper is about how to achieve interoperability in the transformation of humanly usable digital content from one XML content storage system to another. We claim that it will be necessary to use human interpretive intelligence in any such transformation system and that a technology that fails to acknowledge this, and to make suitable provision for it, will be dysfunctional in comparison with one that does. We further claim that a choice about how such content is translated will have implications for whether the form of representation to which the content is transformed, gives access to the meanings which users will wish to extract from it.

Our paper examines three translation systems: Contextual Ontology_X Architecture (the COAX system) and the Common Ground Markup Language, (the CGML system) and Onto-Merge (the Onto-Merge System). All three systems are similar to the extent that they draw upon “merged ontologies” as part of the translation process. However, they differ in the ways in which the systems are designed and how the translation processes work. The CGML and COAX systems are examples of XML to XML to XML based approaches. The OntoMerge system is an example of an XML to Ontology (using web based ontology languages such as OWL) to XML approach.

We think that the leading criterion for selecting any XML translation system is the ease of formulating expanded frameworks/and revised theories of translation that create commensurability between source and destination XML content and merged ontologies. We have discussed this criterion using three sub-criteria: 1) the relative commensurability creation load in using each system, and (2) the system design to integrate the use of human interpretative intelligence and (3) facilities for the use of human interpretative intelligence in content translations. In relation to this decision criterion, we conclude that the CGML system is better than the other two. OntoMerge is next and COAX ranks as the system with the highest commensurability creation load and the least relative provision for incorporating human interpretative intelligence. We note that certain externality effects exist in relation to each of these different systems, including the externalities of initiatives such as the semantic web.

We caution against premature standardisation on any system due to the pressure of externalities.

1 INTRODUCTION

This paper is about how to achieve interoperability in the storage and transfer of humanly usable (HU) digital content; and what must be done to exchange such content between different systems in as automated a way as possible. Automated exchanges of content, within and between industries, help to enable the more general automation of work, in ways which are familiar, say, from on-line banking and from value-chain management. In this paper we are concerned with requirements for the automated exchange of content, when, additionally, this content is designed for human use. As described by Cope and Kalantzis (2004, p 260) interoperability of humanly usable (HU) content allows such content, originally stored in one system for a particular set of purposes, to be transferred to another system and used for a different set of purposes. We are concerned with the design of an efficient converter system, or translator, system, hereafter referred to as a *transformation system*, that makes possible a compatibility of content across different systems.

Our interest in this subject arises because there are substantial sunk costs associated with storing HU content in particular ways. There are also significant costs of translating content from one system to another. These costs will, in due course, become sunk costs also. Therefore, in addressing this topic, our expectation is that, over time, the costs incurred in the storage and transfer of HU digital content will not be regarded as sunk costs, but as investments to secure the continued access to HU digital content, now and into the future.

Many people, who are focused on their own particular purposes, have described *aspects* of the challenges associated with the need for exchanges of HU content. Norman Paskin, the first Director of the Digital Object Identifier (DOI) Foundation, has described difficulties which must be faced when trying to achieve interoperability in content designed for the defense of intellectual property rights¹ (Paskin, 2006). Rightscom (2006, p 40) describe the need to obtain interoperability in the exchange of content, including usage rights, to support the continued development of an e-book industry. McLean and Lynch (2004, p 5) describe the challenges associated facilitating interoperability between library and e-learning systems. The World Wide Web Consortium itself is itself promoting a much larger vision of this same challenge through the semantic web:

¹ These include the need to make interoperability possible across different kinds of media (such as books, serials, audio, audiovisual, software, abstract works, visual material), across different functions (such as cataloguing, discovery, workflow and rights management), across different levels of metadata (from simple to complex), across semantic barriers, and across linguistic barriers.

The semantic web is about two things. It is about common formats for integration and combination of data drawn from diverse sources, where on the original Web mainly concentrated on the interchange of documents. It is also about language for recording how the data relates to real world objects. That allows a person, or a machine, to start off in one database, and then move through an unending set of databases which are connected not by wires but by being about the same thing.

We argue that that there are two, deeper, questions relating to achieving interoperability of HU content that have not been sufficiently discussed. The aim of this paper is to show that these two challenges are interrelated.

First, we claim that it will not be possible to dispense with human intervention in the translation of digital content. A technology that fails to acknowledge this, and to make suitable provision for it, will be dysfunctional in comparison with one that does. The reason for this is that digital content is ascribed meaning by those people who use it; the categories used to organise content reflect these meanings. Different communities of activity ascribe different meanings and thus different categories. Translation of the elements of content from one set of categories to another cannot, we claim, be accomplished without the application of what we will call “human interpretative intelligence”². In what follows, we will provide a detailed explanation of how this is achieved by one translation system, and will argue that any system designed to translate HU content must make this possible.

Second, in order to achieve interoperability of digital content, we claim a choice about how such content is translated will have implications for whether the form of representation to which the content is transformed, gives access to the meanings which users will subsequently wish to extract from it. This has ramifications about the way in which translations are enacted. This, we argue, requires the application of human interpretive intelligence as described above in order that in the translation process, content is mapped between the source and destination schemas.

We advance our general argument in a specific way – by comparing three existing systems which already exist for the abstracting of content to support interoperability. We have used this approach because these three different systems (i) make different assumptions about the use of interpretive intelligence, and (ii) rely on different ontological and technical means for incorporating the use of this

² We particularly acknowledge Paul David’s contribution and thank him for suggesting the use of this term which we have adopted throughout this paper.

intelligence. The first system is the Common Ground Markup Language (or CGML) - whose component ontology has been used to construct the CGML interlanguage. We have termed this the CGML system. The second system is called the Contextual Ontology_X Architecture XML schema whose component ontology is OntologyX. We have termed this the COAX system. The third system is called OntoMerge, whose architecture includes a means of merging and storing different ontologies (a feature referred to as the OntoEngine). We have termed this the OntoMerge System.

The ontological assumptions of these three different systems differ in that CGML system uses noun-to-noun mapping rules to link the underlying digital elements³ of content. In contrast, the COAX system uses verbs and the linkages between digital elements that are generated when verbs are used (as we shall see below it uses what we have termed ‘verb triples’). OntoMerge also uses nouns, but these act as noun-predicates. In a way similar to the COAX system, these noun-predicates also provide linkages between digital elements in the same way as does the use of verb triples in the COAX system. The CGML system is built around the use of human interpretive intelligence, whereas the COAX and OntoMerge systems attempt to economise on the use of such intelligence.

We aim to show that points (i) and (ii) above are related, in a way which has significant implications for how well exchanges of content can be managed in the three systems. We argue that choice of ontological and technical design in the COAX and the OntoMerge systems makes it much harder to apply human interpretive intelligence, and thus it is no surprise that these systems attempt to economise on the use of that intelligence. But we regard the use of such intelligence as necessary. As a result, it is our view that the CGML system would be likely to be more easily implemented than either the COAX or OntoMerge systems.

All three systems are still in their proof-of-concept stage. Because of this, a description of how all these systems abstract content is provided in the body of this paper. We should not be understood as advancing the interests of one system over another. Rather, our purpose in analyzing the systems is to highlight the possibility of locking in to an inefficient standard. We agree with Paul David (2007, p.20) when he suggests that:

preserving open options for a longer period than impatient market agenda would wish is a major part of such general wisdom that history has to offer public policy makers ...

³ We use the word “digital element” in this paper synonymously with the idea of a “digital entity”

The outline of the paper is as follows. In Section 2, we discuss the translation problem and outline a generalized model for understanding the mechanisms for translating digital content. This model is inclusive of two different approaches to the problem, namely, what we have called the XML based inter-language approach and the ontology-based interlanguage approach. In Sections 3 and 4 we discuss these different approaches in some detail, using the three different systems, namely the CGML, COAX and OntoMerge systems as case study illustrations. In Section 5 we provide some criteria for how we might choose any system that addresses the XML translation problem. In Section 6, we highlight some emergent possibilities that might arise based on different scenarios that could develop. And in Section 7 we draw our conclusions.

2 THE TRANSFORMATION OF DIGITAL CONTENT

2.1 A current approach to transformation and interoperability

With current digital workflow practices, translatability/interoperability of content is normally achieved in an unsystematic way by “sticking together” content originally created, so as to comply with separately devised schemas. Interoperability is facilitated manually by the creation of procedures that enable content-transfers to be made between the different end points of digital systems.

For example, two fields expressing similar content might be defined as “manager” in one electronic schema and “supervisor” in another schema. The local middle-ware solution is for an individual programmer to make the judgment that the different tags express similar or the same content and then to manually create a semantic bridge between the two fields – so that “Manager” in one system is linked with “supervisor” in the other system. These middle-ware programming solutions then provide a mechanism for the streaming of data held in the first content storage system categorized under the “manager” field to another content storage system in which the data will be categorized under the “supervisor” field. Such programming fixes make translatability/interoperability possible within firms and institutions and, to a lesser extent, along the supply chains which connect firms, but only to the extent that the “bridges” created by the programmer really do translate two tags with synonymous content. We can call this the “localised patch-and-mend” approach. And we note both that it does involve “human interpretative intelligence,” the intelligence of the programmer formulating the semantic bridges, and also that the use of such intelligence only serves to formulate a model for translation from one schema to another. And as should be plain to everyone, such models can vary in the extent to which they *correctly* translate the content of one schema to another.

Such an approach is likely to be advocated by practical people, for an obvious reason. At the enterprise level the introduction of new forms of information architecture is usually perceived as disruptive of locally constructed approaches to information management. New more systematic approaches, which require changes to content management systems, are often resisted if each local need can be ‘fixed’ by a moderately simple intervention, especially if it is perceived that these more systematic approaches don't incorporate local human interpretative intelligence into information management architecture.

We believe that such patch-and-mend approaches to translatability might be all very well for locally-defined operations. But such localised solutions risk producing outcomes in which the infrastructure required to support the exchange of content within a global economy remains a hopeless jumble of disconnected fixes. This approach will fail to support translatability among conflicting schemas and standards.

2.2 The structure of the automated translation problem

In order to overcome the limitations of the patch and mend approach, we now provide a detailed structure of the problem of creating more generalized translation/transformation systems. This is illustrated in Figure 1. This figure portrays a number of schemas – schemas which refer to different ontological domains. For example, such domains might be payroll, human resources and marketing within an organization; or it might include the domains of publishing, librarianship and bookselling across different sectors.

Our purpose is to discuss the transfer of content from one schema to another - and to examine how translatability/interoperability might be achieved in this process. In Figure 1 we provide details of what we have called the ‘translation/transformation architecture’.

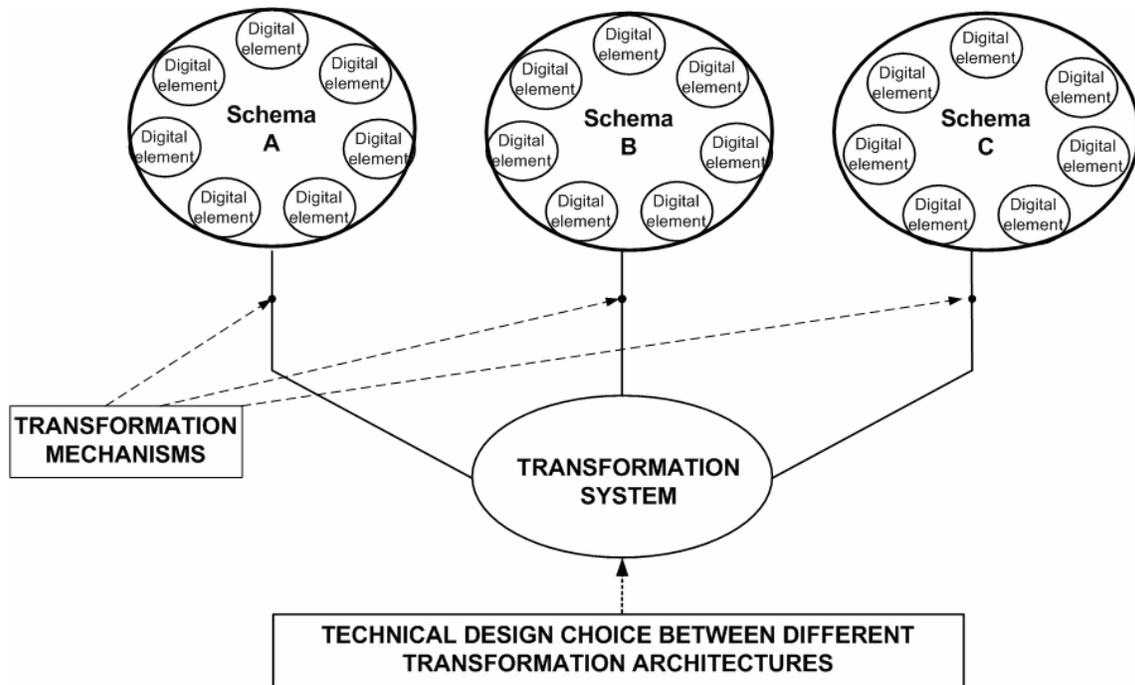


Figure 1. Translation/Transformation Architecture

To consider the details of this translation / transformation architecture, it is first necessary to discuss the nature of the various schemas depicted in Fig 1. XML (Extensible Markup Language) was developed after the widespread adoption of HTML (Hypertext Mark-up Language), which was itself a first step towards achieving a type of interoperability. HTML is a mark up language of a kind that which enables interoperability, but only on a computer-to-computer basis: it enables content to be rendered to a variety of browser-based output devices like laptops and computers. However HTML results only in this in very restricted type of interoperability. XML was specifically designed to address the inadequacies of HTML in ways which make it, effectively, more than a markup language. First, like HTML it uses tags to define the elements of the content in a way that gives meaning to them for computer processing (but unlike HTML there is no limit on the kinds of tags which can be used). This function of XML is often referred to as the semantic function, a term which we will use in what follows. Second, in XML a framework of tags can be used to create structural relationships between other tags. This function of XML is often referred to as the syntactical function, a term which we will also use. In what follows we will call a framework of tags an XML Schema – the Schemas depicted in Figure 1 are simply frameworks of tags.

The extensible nature of XML – the ability to create Schemas using it - gives rise to both its strengths and weaknesses. The strength is that ever-increasing amounts of content are being created to be “XML compliant”. Its weakness is that more and more schemas are being created in which this is done.

In response to this, a variety of industry standards, in the form of XML schemas are being negotiated in many different sectors amongst industry practitioners. These are arising because different industry sectors see the benefit of using the internet as a means of managing their exchanges (e.g. in managing demand-chains and value-networks) because automated processing by computers can add value or greatly reduce labour requirements. In order to reach negotiated agreements about such standards, reviews are undertaken by industry-standards bodies which define, and then describe, the standard in question. These negotiated agreements are then published as XML schemas. Such schemas “express shared vocabularies and allow machines to carry out rules made by people” (World Wide Web Consortium, 2001). The advantage of the process just described is that it allows an industry-standards body to agree upon an XML schema which is sympathetic to the needs of that industry and declares this to be an XML standard for that industry.

But the adoption of a wide range of different XML standards is not sufficient to address the challenge of interoperability in the way we have previously defined this. Obviously reaching negotiated agreements about the content of each XML standard does not address the problem of what to do when content must be transferred between two standards. This problem is obviously more significant, the larger the semantic differences between the ways in which different industry standards handle the same content. This problem is further compounded by the fact that many of these different schemas can be semantically incommensurable.

2.3 Addressing the problem of incommensurability

The problem of incommensurability lies at the heart of the challenges associated with the automated translation problem. Instances of XML schemas are characterized by the use of “digital elements” or “tags” that are used to “mark-up” unstructured documents, providing both structure and semantic content. At face value, it would seem that the translation problem and the problem of incommensurability could be best addressed by implementing a *rule based* tag-to-tag transformation of XML content. The translation “rules” would be agreed upon through negotiated agreements amongst a number of localized stakeholders. This would make this approach similar, but slightly different to the “patch and mend” approach discussed previously, where effectively the arbitrary use of the human

interpretative intelligence of the programmer is used to resolve semantic differences on a tag-to-tag basis.

However, in considering the possibility of either a patch and mend or a rule-based approach, we think there is a necessity to ask the question, *do tags have content* that can be translated from one XML-schema to another, through rule-based tag-to-tag transformations? Or put another way, do such transformations really provide “translations” of “digital content”?

To answer this question we have to be clear about what we mean by “digital content”. This question raises an “ontological” issue in the philosophical, rather than the Information Technology sense of this term. Sharable linguistic formulations about the world (i.e. claims and meta-claims that are speech- or computer- or artifact-based, in other words, cultural information used in learning, thinking, and acting), found in documents and electronic media, have two aspects (Popper, 1972, Ch. 3). The first is the physical pattern of markings or bits and bytes constituting the form of documents or electronic messages embodying the formulations. This physical aspect is comprised of concrete objects and their relationships. In the phrase “digital content,” “digital” refers to the “digital” character of the physical, concrete objects used in linguistic formulations.

The second aspect of such formulations is the pattern of abstract objects that conscious minds can grasp and understand in documents or other linguistic products when they know how to use the language employed in constructing it. The content of a document is the pattern of these abstract objects. It is what is expressed by the linguistic formulation that is the document (ibid.). And when content is expressed using a digital format, rather than in print, or some other medium, that pattern of abstract objects is what we mean by “digital content.”

Content, including digital content, can evoke understanding and a “sense of meaning” in minds, although we cannot know for certain whether the same content presented to different minds evokes precisely the same “understanding” or “sense of meaning.” The same content can be expressed by different concrete objects. For example, different physical copies of Hamlet, with different styles of printing having different physical form, can nevertheless express the same content. The same is true of different physical copies of the American Constitution. In an important sense when we refer to the American Constitution, we do not refer to any particular written copy of it, not even to the original physical document, but rather to the pattern of abstract objects that is the content of the American Constitution and that is embodied in all the different physical copies of it that exist.

Moreover, when we translate the American Constitution, or any other document for that matter, from one natural language -- English -- to another, say German, it is not the physical form that we are trying to translate, to duplicate, or, at least, to approximate in German, but rather it is the content of the American constitution that we are trying to carry or convey across natural languages. This content cannot be translated by mapping letters to letters, or words to words across languages, simply because content doesn't reside in letters or words. Letters and words are the tools we use to create linguistic content. But, they, themselves, in isolation from context, do not constitute content. Instead, it is the abstract pattern of relationships among letters, words, phrases, and other linguistic constructs that constitutes content. It is these same abstract patterns of relationships that give rise to the linguistic context of the assertions expressed as statements in documents, or parts of documents. The notion of linguistic context refers to the language and patterns of relationships that surrounds the assertions contained within documents⁴. We think that it is linguistic context, in this sense of the term, which makes the problem of translation between languages so challenging.

This principle of translation between natural languages is inclusive of XML-based digital content. With XML content it is the abstract patterns of objects in documents or parts of documents combining natural language expressions; relationships among the abstract objects; the XML tags; relationships among the tags, and; relationships among the natural language expressions and the XML tags that gives rise to that aspect of content we call linguistic context.

Once a natural language document is marked up in an XML schema the enhanced document is a metalanguage document, couched in XML, having a formal structure. It is the content of such a metalanguage document, including the aspect of it we have called linguistic context that we seek to translate, when we refer to translating XML-based digital content from one XML language to another.

So, having said what we mean by “digital content,” we now return to the question: do XML tags have content that can be translated from one XML-language to another, through rule-based tag-to-tag transformations? Or put another way, do such transformations really provide “translations” of “digital content”? As a matter of general rule, we think not. And the reason is because of the impact of linguistic context embedded in the content. We contend that this needs to be taken into account, and that tag-to-tag translation systems do not do this.

⁴ The linguistic context is distinct from the social and cultural context associated with any document, because social and cultural context is not necessarily included in the content itself. Rather social and cultural context if it is captured can be included in metadata attached to that content.

Therefore, we think something else is needed to alleviate the growing XML babel by means of a semi-automated translation approach. To address this matter and to set a framework for comparing three different transformation models and the way these different models cater for the application of human interpretative intelligence, we now analyse the details of the translation / transformation architecture outlined in Figure 1.

2.4 System components of the translation / transformation architecture

To address the challenge of incommensurability and linguistic context as described in the previous section, we argue that the design of effective transformation architecture must allow for the application of human interpretative intelligence. The fundamental concern of this paper is to explore the means by which this can be achieved.

Further details of the technical design choice outlined in Figure 1 are now summarised in Figure 2 below.

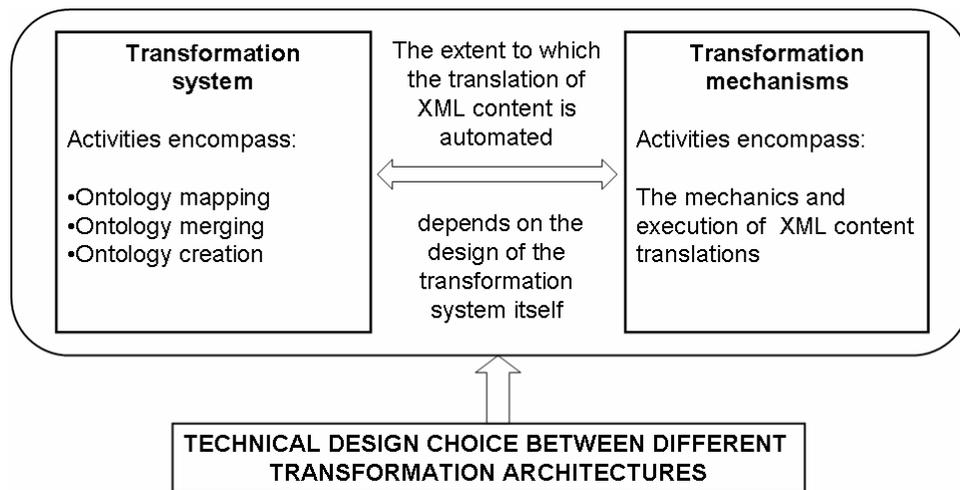


Figure 2: Outline of the technical design choice between different transformation architectures

As a means of exploring aspects of the technical design choice when considering different transformation architectures, it is helpful to understand the ‘translation// transformation system’ as the underlying assumptions and rules governing the translation from one XML schema to another; (including the infrastructure systems that are derived from these assumptions⁵). In turn, the transformation mechanisms are best understood as the ‘semantic bridges,’ or the ‘semantic rules’ that

⁵ Infrastructure systems are discussed in more detail in Section 5.1.

create the connections between each individual digital elements within each XML schema and the translation/transformation system. In this paper we refer to such connections as axioms. We now discuss the details of these concepts in turn.

The translation / transformation system

The design of a translation / transformation system involves the consideration of the means by which the entities that comprise multiple XML schemas are mapped together. This process is called ontology mapping. In this process, we think that in order to integrate incommensurable tags/fields and relations across differing XML schemas, human interpretative intelligence is required to create new ontological categories and hierarchical organizations and reorganizations of such categories. Importantly, when this happens, the ontology mapping process gives rise to the creation of an expanded ontology (ontology creation) – and results in what we will call a “merged ontology”⁶. We call this merged ontology an interlanguage.

The interlanguage approach to translation follows the pattern of the translation/transformation architecture presented in Figure 1. This is because in that architecture, the transformation system is the interlanguage. The interlanguage is comprised of the framework of terms, expressions and rules, which can be used to talk about documents encoded using the different XML schemas that have been mapped into the transformation system.

A key point is that an interlanguage can only apply to schemas whose terms, entities, and relationships have been mapped into it and therefore are modeled by it. The model includes both the relationships among terms in the interlanguage and the mapping relationships between these terms and the terms or other entities in the XML schemas that have been mapped. The model expresses a broader framework that encompasses the ontologies of the XML schemas whose categories and relationships have been mapped into it – thus the term merged ontology.

An interlanguage model will be projectible in varying degrees to XML schemas that have not been mapped into it, provided there is overlap in semantic content between the new schemas and previously

⁶ This idea of an expanded (or merged) ontology has important implications for how to think about the role of categories. We are aware that many might think that the reliance on categories can lead to excessive rigidity in thought processes. However, we think that the use of human interpretative intelligence as a means of addressing incommensurable tags/fields and relations across differing XML schemas addresses this problem. The important point is that in explicitly addressing incommensurability, expanded ontologies with different categories and hierarchies of categories can be and are created according to very specific contexts of application.

mapped schemas. Speaking generally, however, an interlanguage will not be projectible to other schemas without an explicit attempt to map the new schema to the interlanguage. Since the new schema may contain terms and relationships that are not encompassed by the framework (model, theory) that is the interlanguage, it may be, and often is, necessary to add new terms and relationships, as well as new mapping rules to the interlanguage. This cannot be done automatically, and human interpretative intelligence must be used for ontology creation. As we will see when we compare three different systems analysed in this paper, there is potential to make the ontology merging process a semi-automated activity. We shall see that the OntoMerge system is an example of this.

The transformation mechanisms

The transformation mechanisms provide the semantic bridges and semantic rules between individual XML schemas and each translation / transformation system. As we shall see, the design of different transformation mechanisms is shaped by the assumptions / rules governing the transformation system itself. This includes whether human interpretative intelligence is applied during the content translation process itself – or whether the content translation process is fully automated. One of the key themes highlighted in this paper is the extent to which the translation of XML content can occur on a fully automated basis. In the examples we discuss, the COAX and OntoMerge systems are designed to automate the content translation process. In contrast, the CGML system builds within it the capacity to apply human interpretative intelligence as part of the execution of the translation process.

2.5 Conclusion: An interlanguage as a theory of translation

One way to look at the problem of semi-automated translation of XML content is to recognize that an interlanguage is a theory or model of a meta-meta-language. The idea of a meta-meta language arises because each XML schema defines a meta-language which is used to markup text and when different XML schemas are mapped together the framework for cross mapping these schemas becomes a meta-meta language. The interlanguage is supplemented with translation/transformation rules, and therefore is constantly being tested, refuted, and reformulated in the face of new content provided by new XML schemas. This is a good way of looking at things because it focuses our attention on the idea that an interlanguage is a fallible theoretical construct whose success is contingent on its continuing ability to provide a means of interpreting new experiences represented by XML schemas not encountered in the past.

We now turn to two different examples of addressing the translation / transformation challenge: namely the XML-based transformation approach and the ontology-based transformation approach.

3 THE XML-BASED INTERLANGUAGE APPROACH: TWO EXAMPLES

3.1 The XML-based interlanguage approach

In the XML-based interlanguage approach, the transformation system or interlanguage is itself a merged ontology expressed as an XML schema. The schema provides a linguistic framework that can be used to compare and relate different XML schemas. It can also be used along with the transformation mechanisms to translate XML documents marked up in one schema into XML documents using different schemas.

We now supplement this general characterization of XML-based interlanguages with an account of two examples: the Common Ground Markup Language (CGML) and the Contextual Ontology_X Architecture (COAX).

3.2 Introduction to the two examples of XML-based interlanguages

The theoretical foundations of the Common Ground Markup Language (or CGML) system have been written up in Cope and Kalantzis (2004, p 234-270 and WIPO, 2005). CGML forms part of an enterprise publishing system owned by Common Ground Publishing in Australia, called CGpublisher, and a related engine for document transformation called CGexchange (Common Ground Publishing, 2005).

The foundations of the COAX system have been written up in Rightscom (2006). The COAX system forms part of an approach to global infrastructure which is being pioneered by the International Digital Object identifier (DOI) Foundation. It uses an ontology called Ontology_X which is owned by Rightscom in the UK.

The difference in ontological structure between the two systems is visualised in Figure 3.

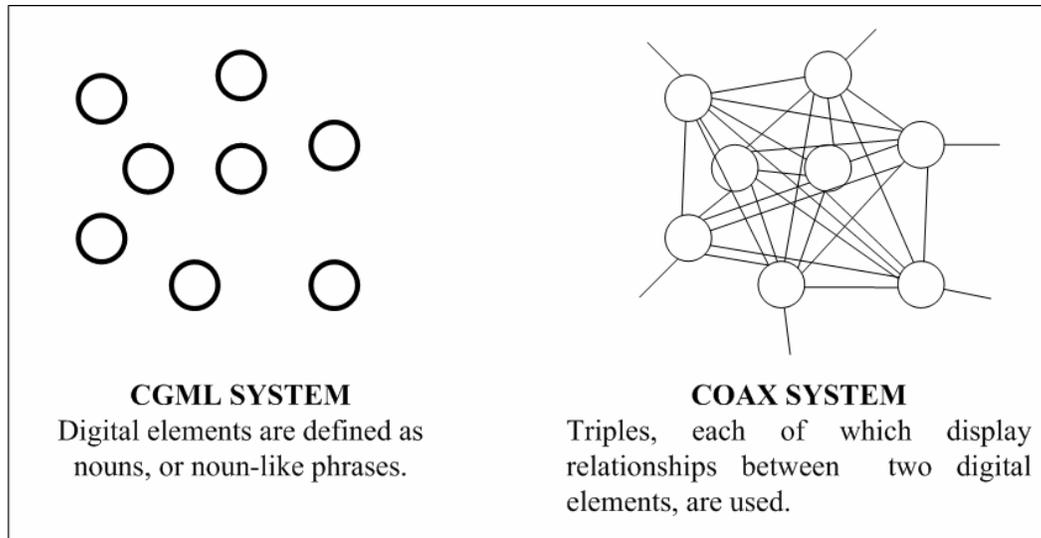


Figure 3: Comparison of the CGML and the COAX systems⁷

We now compare these two systems in some detail, and, in particular, show how the differences in the structures of the merged ontologies impact on the necessity for and the practicality of, human intervention in XML translation processes.

3.3 The CGML translation/transformation architecture

With the CGML system, the digital elements that become part of this transformation system all emanate from the activities of practitioners as expressed in published XML standards. CGML uses semantics and syntax embodied in those standards. All digital tags in the CGML system define nouns. Cope and Kalantzis (2004, p. 248) explain this:

In CGML, all tags are [i.e. define] nouns. CGML, moreover, does not define a noun simply as a kind of word. Rather, CGML tags are defined as lexical items, including pairs or groups of words which in a functional sense combine to form a noun, such as <CopyEditor>.

A key element of the CGML system is the CGML ‘interlanguage’. This is an “apparatus” (ibid, p 259) that is used to describe and translate other languages (Common Ground Publishing, 2005). In particular, the CGML application provides a transformation system through which the digital elements expressed in one XML standard can be transformed and expressed in another standard. The interlanguage is governed by two “semantic logics” (Cope and Kalantzis, 2004, p. 251).

⁷ The right hand side of this Figure is referenced to the Universal Data Model (Rust, 2005, slide 24)

The first is that of *distinction-exclusion*. This helps identify tags whose meanings exist as parallel branches (sibling relations) – i.e. those tags which have synonymous meanings across different standards and, by implication, those that do not⁸. The second logic is that of *relation-inclusion*. This determines tags contained within sub-branches (parent-child relations) which are semantically included as part of the semantics of the superordinate branch⁹ (ibid, p 251).

The impact of these two logics gives rise to the semantic and the syntactical rules that are embedded within the CGML tag dictionary and the CGML thesaurus. Cope and Kalantzis explain the role of both the CGML dictionary and thesaurus in this way.

The CGML tag thesaurus takes each tagging schema as its starting point, lists its tags and reproduces the definitions and examples as given by each tagging schema. Against each tag, a direct CGML synonym is provided, whose semantics are coextensive with, or narrower than, the tag against which the mapping occurs. (ibid, p 252).

The CGML tag dictionary links the tag concept to a semantically explicit definition¹⁰. The building blocks of the CGML Dictionary are the other tag-concepts of the CGML interlanguage, (ibid, p 256).

A rule of “lowest common denominator semantics” is rigorously applied to the CGML tag dictionary (ibid, p 258). Obviously, the contents of the thesaurus and the dictionary can be extended, each time a new XML standard is mapped into the CGML interlanguage.

By systematically building upon the two logics described above, Cope and Kalantzis (2004) explain the interlanguage apparatus in this way.

The interlanguage mechanism does not manage structure and semantics per se; rather, it automatically manages the structure and semantics of structure and semantics. Its mechanism,

⁸ For example, a <Person> is not an <Organisation> because an <Organisation> is defined as a legally or conventionally constituted group of <Persons> (Cope and Kalantzis, 2004, p 251).

⁹ For example, a <Party> to a creative or contractual relationship can be either a <Person> or an <Organisation> (Cope and Kalantzis, 2004, p 251).

¹⁰ The CGML dictionary does not purport to be about external referents as ‘meaning’; rather, it is built via the interlanguage technique from other languages which purport to have external referents. As a consequence, its meanings are not the commonsense meanings of the life-world of everyday experience, but derivative of specialised social languages. As such, CGML dictionary is more like a glossary than a normal dictionary. (Adapted from Cope and Kalantzis 2004, p. 255).

in other words, is meta-structural and meta-semantic. It is aimed at interoperability of schemas which purport to describe the world rather than immediate reference to the world. We have named its underlying mechanism the ‘interlanguage’ apparatus (ibid, p 259).

The effect of these crosscutting processes is the systematic mapping of existing and emerging tagging schemas [standards]¹¹ against each other (ibid, p. 253).

We note that CGML is only in an early stage of development. In the future, we would understand that the CGML interlanguage apparatus could interpret the semantics and syntax of content expressed in any XML schema – and not just content represented in an XML standard. We make this point, because it has some implications for comparing the CGML system with the use of surface and standard ontologies within the OntoMerge system discussed in Section 4.

3.4 How the CGML system works

Cope and Kalantzis (2004, p 263) explain how the CGML system works. The key requirement for it to work on a particular XML document or content is that the XML standard underlying the document has already been mapped into the CGML interlanguage. The mapping process requires a comparison of the noun tags in the XML standard with the nouns in the existing CGML ontology at the time of the mapping. This process requires the application of human interpretative intelligence. Generally speaking, there will be considerable overlap in noun tags between the CGML system and the XML standard; on the average there may be as much as 70% “common ground” between them. For the remaining 30% however, it will be necessary to construct transformation rules that explicitly map the noun tags in the standard to noun tags in CGML. Where this can't be done, new noun tags are taken from the standard and added to CGML, which means placing the new tags in the context of the hierarchical taxonomy of tags that is CGML, and also making explicit the translation rules between the tags in the standard and the new tags that have been added to the CGML system.

An overview of the resulting revised CGML translation/transformation system, as it is applied to XML content, is provided in Figure 4. We can describe how this system works on content as a two-stage process.

¹¹ We have inserted the brackets

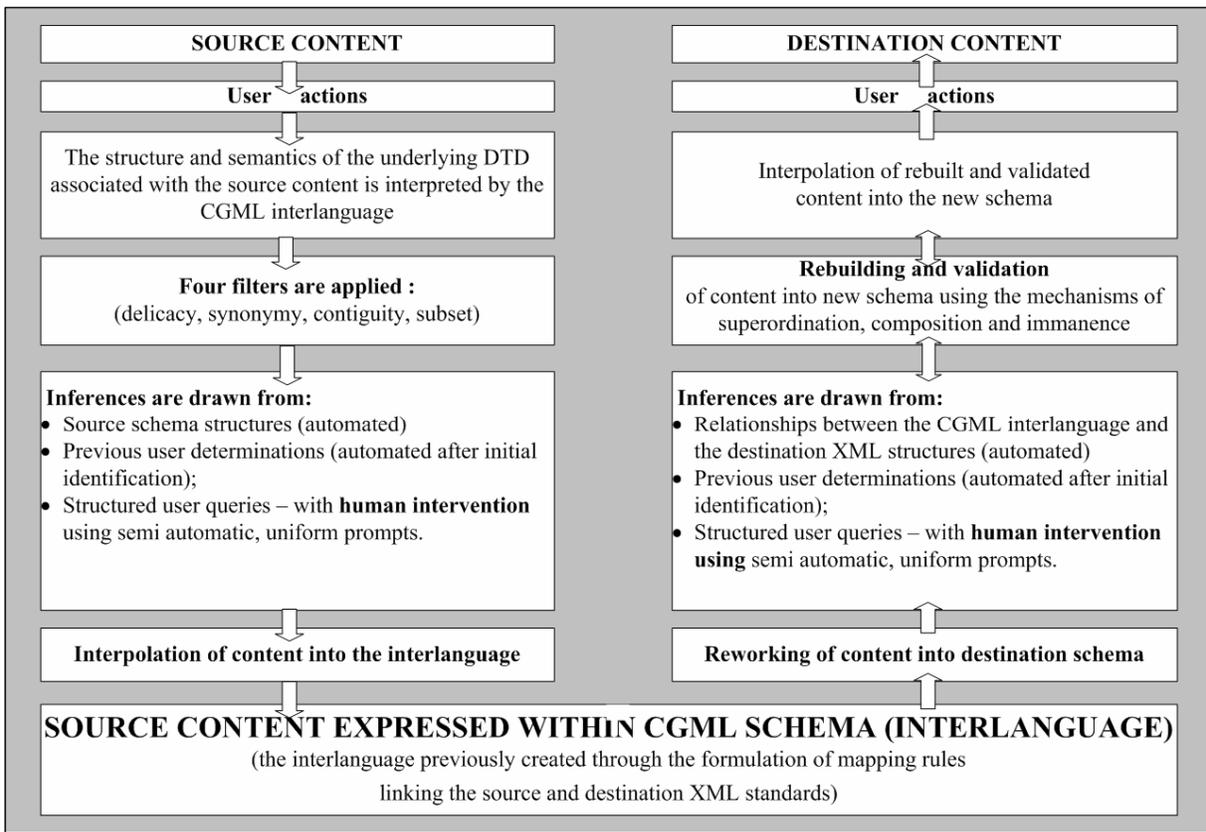


Figure 4. XML Content Translation Using the CGML Transformation System

Stage 1: Merging of the source XML content with the CGML XML schema

To begin with, the CGML interlanguage apparatus interprets the semantics and syntax of the source content. It is able to do this from the underlying XML standard¹² and the particular way the source content is expressed within the semantic and syntactic structure of the standard itself. For this to be possible, it is, once again, necessary that each XML standard has already been mapped into the CGML interlanguage.

The translation of the source content into the CGML interlanguage is then carried out by passing the source content through four different filters.

¹² In the future, in principle, this same approach could be used as an “ontology reader”. That is, the interlanguage apparatus could interpret the semantics and syntax of content expressed in any XML schema – and not just content represented in an XML standard.

- A **delicacy filter** determines whether an aggregated content element needs to be disaggregated and re-tagged. This ensures that content is tagged at its lowest level of granularity within the CGML schema itself.
- A **synonymy filter** identifies any content elements where ambiguity of meaning exists.
- A **contiguity filter** is used to determine the “relationship proximity” of different content elements associated with the source and the destination standards. This filter draws upon three principles associated with the structuring of taxonomies – the mechanisms of superordination, composition and immanence¹³.
- A **subset filter** determines at what level in the hierarchy of composition or superordination content needs to be entered to effect superordinate transformations.

The translation can occur in automated ways, when inferences can be drawn from the underlying XML standard. Where it is not possible to construe any direct relationships between content elements from the source standard and the interlanguage, a structured query is thrown up for the user to respond to. It is in this process that human interpretative intelligence is applied to enable translation. User responses to these structured queries become part of the accumulated “recordings” of semantic translations. These responses are then built up into a ‘bank’ of previous translations. The results stored in this bank can be used to refine the operation of the filters¹⁴ so that, as the transformation process is repeated, fewer and fewer structured queries are thrown up for the user to respond to. The user responses grow the knowledge about translation contained in the translation system, and also grow the ontology resident in CGML. This shows that “human interpretative intelligence” is about solving problems that appear in the translation system, and raises the question of whether the CGML system might have an advantage over other XML-based systems that do not provide for human intervention in problem solving and the growth of translation knowledge.

¹³ **Superordination** relations perform the function of sub-classification. They express an ‘is a’ relationship between one level in the taxonomic hierarchy and another. <Book> is a <Product>, as is an <AudioRecording>. **Composition** relations, by contrast, connect parts into wholes. They express a ‘has a’ relation between levels in the taxonomic hierarchy. A <GlossaryItem> and a <GlossaryItemDefinition> are both parts of a <Glossary>. <Glossary> is not functional without both of these parts. **Immanence** expresses an ‘underlies’ relationship between contiguous levels in the taxonomic hierarchy. A <Design> underlies a <Work> and a <Work> underlies a <Product>. (Cope and Kalantzis, 2004, p. 251).

¹⁴ It is possible, in principle, for the filters to be ‘trained’, in a similar way to that in which voice recognition software is trained to recognize the speech of particular speakers.

Stage 2: Transformation of CGML XML content into the destination standard

Once the content is interpolated into the CGML interlanguage, the content is structured at a sufficient level of delicacy to enable the CGML transformation system to function. Some re-configuration of the content is necessary so that appropriate digital elements can be represented according to the semantic and syntactic structures of the destination XML standard. This process involves passing the content ‘backwards’ through only two of the filters but with the backwards filter constraints set according to the requirements of the destination standard. Only the contiguity and subset filters are required because when content is structured within the CGML interlanguage apparatus, the content already exists at its lowest level of delicacy and semantic composition and thus the delicacy and synonymy filters are not needed. The three mechanisms associated with this backwards filtering process are superordination, composition and immanence – as outlined in the footnote below. A number of “sub-mechanisms” are associated with this process.¹⁵

This translation can also occur in automated ways when inferences can be drawn from the destination standard. In the case where the filter constraints prevent an automated passing of content, then a structured query is thrown up for the user to respond to, as has been described previously. User responses are then built up into a bank of previous translations. The results stored in this bank can also be used to refine the operation of the filters so that, as the translation process is repeated, fewer and fewer structured queries are thrown up for the user to respond to in this second stage as well as in the first stage.

¹⁵ The sub-mechanisms associated with superordination are as follows.

- *Hyponymy*. The principle of classification is *includes in its class*.
- *Hyperonymy*. The principle of classification is *is a class of*.
- *Co-hyperonymy*. The principle of classification is *is the same as*.
- *Antonymy*. The principle of classification is *is the converse of* and;
- *Series*. The principle of classification is *is related by gradable opposition to*.

The sub-mechanisms associated with composition are as follows.

- *Meronymy*. The principle of classification is *is the same as*.
- *Co-meronymy*. The principle of classification is *is integrally related to but exclusive of*.
- *Consistency*. The principle of classification is *is made of*.
- *Collectivity*. The principle of classification is *consists of*.

3.5 The COAX translation / transformation architecture

The COAX transformation system differs from the CGML system. This difference comes from the importance that verbs play in the way in which the COAX system manages semantics and syntax.

...Verbs are the most influential terms in the ontology, and nouns, adjectives and linking terms such as relators all derive their meanings, ultimately, from contexts and the verbs that characterize them. (Rightscom, 2006, p 46).

The reason for this is given by Rightscom (ibid, p 46) as follows:

COA semantics are based on the principle that meaning is derived from the particular functions which entities fulfil in contexts. An entity retains its distinct identity across any number of contexts, but its attributes and roles (and therefore its classifications) change according to the contexts in which it occurs.

The semantics and syntactical transformations associated with the COAX system are dependent upon all tags within the source standard being mapped against an equivalent term in the COAX system (this is explained in more detail in Section 3.6). By working in this way, Rust (2005, slide 13) suggests that the COAX system can be understood as an *ontology of ontologies*.

Rightscom (ibid. p. 27) explains the process by which an XML source standard is mapped into COAX this way:

For each schema a once-off specific “COA mapping” is made, using *Ontology_X*. This mapping is made in “triples”, and it represents both the syntax and semantics of the schema. For example, it not only contains the syntactic information that element X is called “Author” and has a Datatype of “String” and Cardinality of “1-n”, but it contains the semantic information that “X IsNameOf Y” and that “Y IsAuthorOf Z”. It is this latter dimension which is unusual and distinguishes the COAX approach from more simple syntactic mappings which do not make the semantics explicit.

3.6 How the COAX Transformation System Works

What follows is an explanation of how the COAX transformation system works. This material has been adapted from Rightscom (2006). We can also present this as a two staged process.

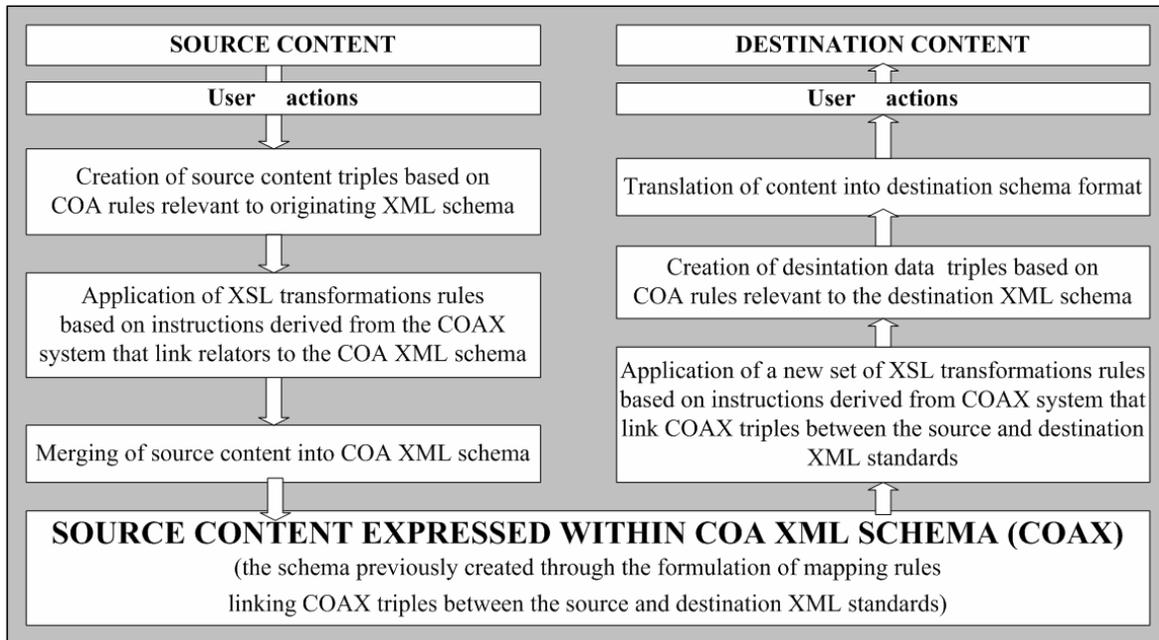


Figure 5. XML Content Translation Using the COAX Transformation System

Stage 1: Merging of the source XML content with the COA XML schema

To begin with, all “elements” from the source content are assigned ‘relators’ using a series of rules which are applied using the COAX transformation system. A ‘relator’ is an abstraction expressing a relationship between two elements - the domain and range of the relator (ibid, p. 57). The relators may be distinct for each different XML standard and all originate from the COAX dictionary. The relators and the rules for assigning relators will have been created when each XML standard is mapped into the COAX system. Obviously for a user to be able to proceed in this way, it is necessary that each XML standard has already been mapped into the COAX system.

The linked elements created by the assigning of relators are expressed as XML structured triples¹⁶. “The COA model is “triple” based, because content is expressed as sets of domain-relator-range

¹⁶ “The COA model is “triple” based, because content is expressed as sets of Domain-Relator-Range statements such as “A HasAuthor B” or “A IsA EBook”. (Rightscom, 2006, p. 25). This domain-relator-range is in compliance with the W3C RDF standard subject-predicate-object triple model (ibid, p 16).

statements such as “A HasAuthor B” or “A IsA EBook” (ibid, p 26). These triples contain the semantic and syntactic information which is required for content transformation. Notice that relators must be constructed to express the relationship between every pair of digital elements if these elements are to be used within the COAX transformation system. The aggregation of triples form the basis of the COAX data dictionary Data dictionary items are created when the elements in the source and destination XML schemas are mapped against each using relators as the basis for mapping COAX triples.

Once the content is specified as a set of triples, then the COAX system uses the Extensible Stylesheet Language, or XSL¹⁷ to further transform the source content, now expressed as triples, into a form whose semantics and syntax are compliant with the COAX schema. The XSL transformations, or (XSLTs), which are used to do this contain within them rules generated by Rightscom that are specific to the transformation of the content from the set of triples into the COAX interlanguage. For a user to be able to do this requires that the necessary rules and XSLT instructions have been mapped into the COAX system. Unlike the CGML XML content translation process, the COAX process of merging XML content into COAX is completely automatic and there is no facility for user queries to determine how content not represented in COAX may be integrated into the merged COAX ontology. Thus, the use of human interpretative intelligence in COAX is limited to the contributions of Rightscom experts who create the bridge mappings of XML standards to COAX.

Stage 2: Transformation of the COAX content into the destination standard

Once the content is structured according to the COAX schema requirements, a new set of XSLTs are applied to the content in order to transfer the content from the COAX schema format into a set of triples which contain the semantic and syntactic information which is needed to convert the content into a form compatible with the destination XML standard. The new XSLTs used to do this, also contain within them rules generated by Rightscom, and are specific to the transformation of COAX-formatted content to the destination standard. Again, for a user to be able to do this it is necessary that each XML standard has already been mapped into the COAX system.

¹⁷ XSL is an XML-based language which is used for the transformation of XML documents (Wikipedia, 2006)

A further series of rules are then applied using the COAX transformation system to translate the content, expressed as a set of triples, into the format of the destination XML standard. These rules will be different for each destination XML standard and are created when each XML standard is mapped into the COAX system. Again, for a user to be able to proceed in this way, it is necessary that the destination XML standard has already been mapped into the COAX system.

4 THE ONTOLOGY-BASED INTERLANGUAGE APPROACH: ONTOMERGE

4.1 The Ontology-based interlanguage approach using OWL

In the Ontology-based interlanguage approach, the transformation system or interlanguage (see Figure 1) is different from what we have described with CGML and COAX. This difference in approach arises because an ontology-based interlanguage approach does not result in an XML to XML to XML translation. Rather, it is an XML to Ontology to XML translation. Like in the XML-based interlanguage approach, the transformation system is itself a merged ontology. But unlike the XML-based interlanguage approach the merged ontology is not expressed as an XML schema. In contrast, it is expressed as a semantic web ontology through the use of object-oriented modeling languages such as the Web Ontology Language (OWL). OWL is designed for use by applications that need to process the content of information instead of just presenting information to humans (World Wide Web Consortium, 2004a). Processing the content of information includes the ability to lodge queries across differing information systems that might be structured using different schema frameworks.

In achieving the objective of “processing content” these merged ontologies, in turn can be used in two ways. First they can assist with the translation of different ontologies through the continuous expansion of the transformation system to create larger more expansive merged ontologies. Second, they can be used in the translation of one XML document to another.

This highlights that XML schemas and ontologies are not the same thing. An XML schema is a language for restricting the structure of XML documents and is concerned with the message format associated with a particular knowledge representation (World Wide Web Consortium, 2004b). In contrast ontology represents the meaning of terms in vocabularies and the inter-relationships between those terms (World Wide Web Consortium, 2004a).

But despite the differences in the way the inter-language mechanisms work, the overall system architecture is the same as that outlined in Section 2.2. That is, the transformation system is the interlanguage, but the interlanguage is expressed as a semantic web ontology using object orientated modeling languages. The ontology comprises a framework of terms, expressions and rules which can be used as a basis for analysing documents encoded using the different XML schema that have been mapped into the transformation system and for supporting queries between different information systems that have been structured across different ontologies.

To work through the differences in the ways the semantic web interlanguage systems work, we now turn to one example of such a translation / transformation architecture, namely OntoMerge.

4.2 OntoMerge: An Example of an Ontology-based Translation System

OntoMerge is an online service for Ontology Translation, developed by Drew McDermott, Dejing Dou, and Peishen Qui (2004) located at Yale University in the United States. It is an example of a translation / transformation architecture that is consistent with the design principles of the semantic web. Some of the design principles of the semantic web that form part of the OntoMerge approach include the use of formal specifications such as Resource Description Framework (RDF)¹⁸, OWL and the predecessor language such as DARPA Agent Markup Language¹⁹ (DAML) and the Ontology Inference Layer (OIL). To develop OntoMerge, the developers have also built their own tools such as PDDAML²⁰. PDDAML has been designed to do translations between the Planning Domain Definition Language (PDDL) and DAML.

Specifically, OntoMerge:

. . . serves as a semi-automated nexus for agents and humans to find ways of coping with notational differences between ontologies with overlapping subject areas. OntoMerge is developed on top of PDDAML (PDDL-DAML Translator) and OntoEngine (inference engine).

“OntoMerge accepts (a) a set of concepts or instance data based on one or more DAML

¹⁸ The Resource Description Framework (RDF) is a general-purpose language for representing information in the Web (World Wide Web Consortium, 2004)

¹⁹ The objective of DAML is to develop a language and tools to facilitate the concept of the Semantic Web

²⁰ See http://www.cs.yale.edu/homes/dvm/daml/pddl_daml_translator1.html

ontologies (b) a target ontology; and produces the concepts or instance data translated to the target ontology.”

More recently, OntoMerge has acquired the capability to accept DAML + OIL²¹ and OWL ontologies, as well. But for it to be functional, OntoMerge requires merged ontologies in its library. These merged ontologies specify relationships among terms from different ontologies.

OntoMerge relies heavily on *Web-PDDL*, a strongly typed, first-order logic language, as its internal representation language. Web-PDDL is used to describe axioms, facts, and queries. It also includes a software system called *OntoEngine*, which is optimized for the ontology-translation task” (2004, p. 6). Ontology translation may be divided into three parts. First, there is syntactic translation from the source ontology expressed in a web language, to an internal representation, e.g., syntactic translation from an XML language to an internal representation in Web-PDDL. Second, there is semantic translation using this internal representation. This translation is implemented using the merged ontology derived from the source and destination ontologies, and the inference engine to perform formal inference. Third, there is syntactic translation from the internal representation to the destination web language. In doing the syntactic translations, there's also a need to translate between Web-PDDL and OWL, DAML or DAML + OIL. OntoMerge uses its translator system PDDAML to do these translations.

“Ontology merging is the process of taking the union of the concepts of source and target ontologies together and adding the bridging axioms to express the relationship (mappings) of the concepts in one ontology to the concepts in the other. Such axioms can express both simple and complicated semantic mappings between concepts of the source and target ontologies.” (Dou, McDermott, and Qi, 2004a, pp. 7-8)

“. . . Assuming that a merged ontology exists, located typically at some URL, OntoEngine tries to load it in. Then it loads the dataset (facts) in and does forward chaining with the bridging axioms, until no new facts in the target ontology are generated.” (ibid, 2004a, p. 12).

²¹ Like OWL, DAML+OIL is a semantic markup language for Web resources. It builds on earlier W3C standards such as RDF and RDF Schema, and extends these languages with richer modelling primitives (World Wide Consortium, 2001)

Merged ontologies created for OntoMerge act as a “bridge” between related ontologies. However, they also serve as new ontologies in their own right and can be used for further merging to create merged ontologies of broader and more general scope.

Ontology merging requires human interpretative intelligence to accomplish it successfully, because ontology experts are needed to construct the necessary bridging axioms, (or mapping terms) from both the source and destination ontologies. Sometimes, also, new terms may have to be added to create bridging axioms, and this is another reason why merged ontologies have to be created from their component ontologies. A merged ontology contains all the terms of its components, as well as any new terms that were added in constructing the bridging axioms.

Dou, McDermott, and Qi (2004b, p. 14) themselves emphasize heavily the role of human interpretative intelligence in creating bridging axioms.

In many cases, only humans can understand the complicated relationships that can hold between the mapped concepts. Generating these axioms must involve participation from humans, especially domain experts. You can't write bridging axioms between two medical-informatics ontologies without help from biologists. The generation of an axiom will often be an interactive process. Domain experts keep on editing the axiom till they are satisfied with the relation expressed by it. Unfortunately, domain experts are usually not very good at the formal logic syntax that we use for the axioms. It is necessary for the axiom-generating tool to hide the logic behind the scenes whenever possible. Then domain experts can check and revise the axioms using the formalism they are familiar with, or even using natural-language expressions.

4.3 OntoMerge and the translation of XML content

It is in considering the problem of XML translations a distinguishing feature of the OntoMerge system is revealed as compared to the CGML and COAX approaches. OntoMerge is not reliant on the declaration of XML standards in order for its transformation architecture to be developed. This reflects OntoMerge's semantic web origins and the objective of “processing the content of information”. Because of this, the OntoMerge architecture has developed as a “bottom up” approach to content translation. We say this because with OntoMerge, with the translation of XML content occurs there is no need to reference declared in XML standards in the OntoMerge transformation architecture. Nor is

there any assumption that content needs to be XML standards based to enact successful translation. In other words, OntoMerge is designed to start at the level of content and work upwards towards effective translation. We say work upwards, because as we explain later, OntoMerge is reliant upon access to semantic web ontologies to execute successful translations.

But even despite this, with the OntoMerge system architecture, there is a need to distinguish between a “surface ontology” and a “standard (or deep) ontology”. A surface ontology is an internal representation of the ontology derived from the source XML content when the OntoEngine inference engine is applied to the source content. In contrast, a standard ontology focuses on domain knowledge and thus is independent of the original XML specifications (Qi et al, 2004, p 7)²².

Thus surface and standard ontologies in the ontology-based interlanguage approach appear to be equivalent to XML schemas and standards in the XML-based interlanguage approach. So, where as with the XML-based interlanguage approach, there is a reliance on declared XML standards, with the ontology-based interlanguage approach there is a need to draw upon published libraries of standard ontologies or semantic web ontologies.

In translating XML content using the OntoMerge system architecture the dataset is merged with the surface ontology into what we have called a surface ontology dataset. In turn, this is subsequently merged with the standard ontology to create what we have termed a standard ontology dataset²³. Once the content is expressed in the standard ontology²⁴ dataset it can then be translated into a standard ontology dataset related to the destination content schema.

In executing such translations of XML content, the OntoMerge system also uses bridging axioms. These are required in the translation between different standard ontology datasets. Thus with the OntoMerge system bridging axioms act as the transformation mechanisms and are central to the translation of XML content (ibid, p 13). With both the CGML and COAX systems, the transformation mechanisms play a similar role to that of the bridging axioms in OntoMerge. In CGML, we saw that rules associating noun to noun mappings were critical to performing translations. For COAX, we saw that COAX triple to COAX triple mappings were the most influential terms in the ontology. In OntoMerge, the key to translation are the bridging axioms that map predicates to predicates in the

²² Qi et al (2004, p 7) highlight that the role of the standard ontology is contrary to the surface ontology, they call it a deep ontology.

²³ The details of how this merging takes place are described in the following section.

²⁴ The choice of the standard ontology is determined by its relatedness to the nature of the source content

source and destination standard ontologies respectively.

Predicates are terms that relate subjects and objects and are central to RDF and RDF triples. In RDF, a resource or subject (or name) is related to a value or object by a property. The property, the predicate (or relator) expresses the nature of the relationship between the subject and the object (or the name and the value). The assertion of an RDF triple says that some relationship indicated by the triple, holds between the things denoted by the subject and the object of the triple (World Wide Web Consortium, 2004c).

Examples of predicates include: “author,” “creator,” “personal data,” “contact information,” “publisher.” Some examples of statements using predicates are:

- 1) Routledge is the ***publisher*** of All Life Is Problem Solving
- 2) John's ***postal address*** is 16 Waverly Place, Seattle Washington
- 3) Michelangelo is the ***creator*** of David

With OntoMerge then, predicates are nouns, and bridging axioms are mapping rules that use predicates to map nouns-to-nouns as in CGML. However, there are significant differences in approach because these nouns as predicates carry with them the inferred relationships between the subjects and objects that define them as predicates in the first place. This is the case even though the bridging axioms are not defined using triples mapped to triples as in the case of COAX.

The importance of predicates is fundamental to the OntoMerge inference engine, OntoEngine, because one of the ways of discriminating “the facts” embedded within datasets loaded into OntoEngine is through the use of predicates. That is, predicates forms the second tier of indexing structure of OntoEngine (Dou et al, 2004b p 8). This in turn provides the foundation for OntoEngine to undertake automated reasoning (ibid, p 8).

When some dataset in one or several source ontologies are input, OntoEngine can do inference in the merged ontology, projecting the resulting conclusions into one of several target ontologies automatically.

4.4 How the OntoMerge system works

There are some differences in the way OntoMerge is used to translate XML content versus how it is applied to translate ontologies. The general approach taken to XML content translation has been outlined by Dou, McDermott, and Qi (2004 b, p 5) as follows:

We can now think of dataset translation this way: Take the dataset and treat it as being in the merged ontology covering the source and target. Draw conclusions from it. The bridging axioms make it possible to draw conclusions from premises some of which come from the source and some from the target, or to draw target-vocabulary conclusions from source-language premises, or vice versa. The inference process stops with conclusions whose symbols come entirely from the target vocabulary; we call these target conclusions. Other conclusions are used for further inference. In the end, only the target conclusions are retained; we call this projecting the conclusions into the target ontology. In some cases, backward chaining would be more economical than the forward-chain/project process . . . In either case, the idea is to push inferences through the pattern.

The details of this approach to translating XML documents into other XML standards are outlined in Figure 6. As with the CGML and COAX approaches, we have also described how this system works as a two-staged process.

Stage 1: Transformation of source content schema into the merged ontology data set

To begin with the OntoEngine inference engine is used to automatically build a surface ontology from the source XML content's Document Type Definition (DTD) file²⁵. The surface ontology is the internal representation of the ontology derived from the source DTD file and is expressed using Web-PDDL (mentioned previously). Drawing upon the vocabulary of the surface ontology, OntoEngine is used to automatically extract a dataset from the original XML content and the surface ontology. Though Qi et al (2004, p 5) don't use this term, we call the new dataset, the source document surface dataset.

²⁵ A Document Type Definition (DTD) defines the legal building blocks of an XML document. It defines the document structure with a list of legal elements and attributes (W3 Schools, viewed on 16 Aug, 2008)

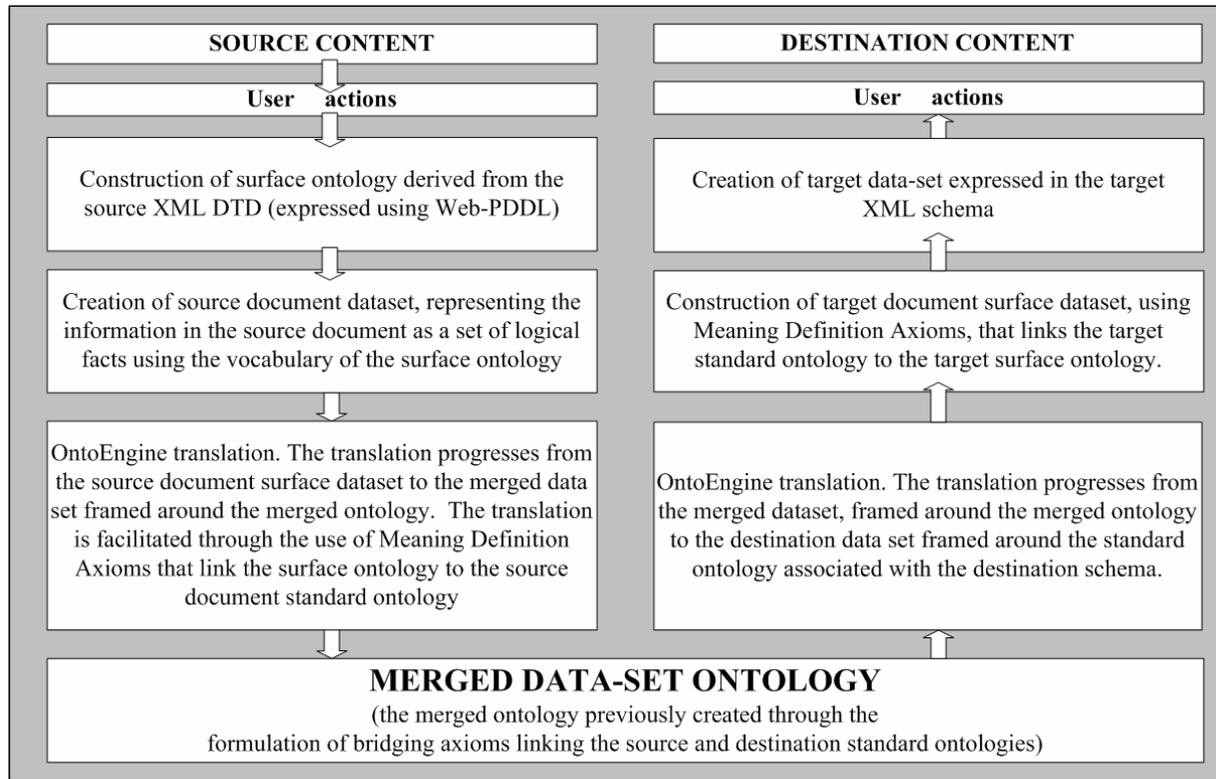


Figure 6. XML Content Translation Using OntoMerge

This dataset represents the information in the source document “as a set of logical facts using the vocabulary of the surface ontology.” The need then is to further merge this dataset into a standard ontology dataset. In order to accomplish this merging, there is a requirement that the surface ontology has been merged with the standard ontology through the formulation of different types of bridging axioms called Meaning Definition Axioms - MDAs (ibid, p 2). According to Qi et al MDAs are required to assign meaning to the source XML content. This is achieved by relating the surface ontology to the particular standard ontology. Like in the case of formulating bridging axioms, these MDAs are created with the help of human interpretative intelligence. Once these MDAs are formulated, the merging process can proceed with an automatic translation of the source document surface dataset to a dataset expressed in the standard ontology. The translation is also undertaken using the OntoEngine inference engine. The translation is able to proceed because of inferences derived from logical facts that are accessible from the use of MDAs that link the surface ontology-data set to the standard ontology. We now call this the source document merged ontology dataset.

Stage 2: Transformation of the source document merged ontology dataset into the destination XML schema

The source document merged ontology dataset is then translated to the destination document merged ontology. This translation is executed using OntoEngine and once again this translation is conditional upon the formulation of bridging axioms that link the source and destination standard ontologies. The translation results in the creation of a destination document merged ontology dataset. In turn, this is translated into destination document surface ontology dataset also using OntoEngine. This translation is conditional upon MDAs being formulated that link the destination standard ontology with the destination surface ontology. The destination document surface ontology dataset is subsequently used along with the DTD file for the destination XML standard or schema, to create a destination dataset expressed in the destination XML standard (or schema). This completes the translation.

4.5 Differences in approach

As we have explained previously, the OntoMerge approach to XML translation is different from the CGML and COAX approaches in that it is pursued in a much more bottom up, rather than top-down manner. That is, when someone is faced with an XML translation problem and wants to use the OntoMerge approach, they don't look to translate their surface ontologies to a single interlanguage model as represented by a model like the CGML or COAX systems. Rather, they first have to search web libraries to try to find a standard ontology (written in DAML, DAML + OIL, or OWL) closely related to the knowledge domain of the surface ontology that has been merged with the surface ontology through the formulation of MDAs. Secondly, they also need to search web libraries for an earlier merged ontology that already contains bridging axioms linking the source standard ontology and the destination standard ontology. Thirdly, they also need to search web libraries for previously developed ontologies that merge the destination standard ontology with the destination surface ontology, through the formulation of meaning definition axioms.

If no such merged ontology exists, then no translation can be undertaken until domain experts working with OntoMerge can construct MDAs mapping the predicates in their surface ontologies to an ontology already available in DAML, DAML + OIL, or OWL ontology libraries. Thus far, there appears to be no standard deep ontological framework that merges all the ontologies that have been developed in

OntoMerge application work. There are many islands in the OntoMerge stream. But the centralized integration offered by CGML and COAX is absent.

5 CHOOSING AMONG SYSTEMS

In previous sections we have discussed the general nature of the transformation/translation problem from digital content expressed in one XML schema to the same digital content expressed in another. We have then reviewed three systems for performing such translations. Two of these systems originated in the world of XML itself and aim to transform the ad hoc “patch-and-mend” and rule based tag-to-tag approaches to transformation with something that would be more automatic, more efficient, and equally effective. The third system originated in the world of the semantic web as a solution to the need for a software tool that would assist in performing ontology translations. The OntoMerge system was developed to solve this generalized problem and was found to be applicable to the problem of XML schema translations. As we've seen previously, all three systems rely on a merged ontology approach that creates either a generalized interlanguage, or in the case of OntoMerge, a less generalized non-XML-based interlanguage that will work to translate one XML document to another.

In this section we confront the problem of system choice. How do we select among competing architectures designed to enable XML translations? System selection, of course, requires one or more criteria which we now proceed to outline and discuss. Then we illustrate the application of the criterion or criteria to the three illustrative systems reviewed earlier.

A generalized criterion for selecting among systems arises from the problem of semantic incommensurability that arise during the process of translating XML content between different schemas or standards. It is the criterion of ease of creating such commensurability (Popper, 1970, pp. 56-57, Firestone and McElroy, 2003, p. 161) where needed, so that transformations of XML content between one schema (or standard) and another, can be implemented. In general, the system supporting the greatest relative ease and convenience in creating commensurability is to be preferred.

Another way to look at this arises from the recognition that the merged ontologies used in all three approaches may be viewed as fallible theories of transformation/translation among variant XML standards and schemas. Each time a dataset representing a new standard or schema is encountered that cannot be translated using a previously developed merged ontology, the theory of translation using that ontology is falsified by the new data that has been encountered, and there is a need to revise that theory

of translation by creating a revised merged ontology whether in CGML, COAX, OntoMerge, or some other system we have not yet reviewed.

Thus, to use such transformation/translation architectures, there is a continuing need to reformulate the embedded theories of translation and merged ontologies. Again, the system that makes it easiest to formulate expanded frameworks/arising from revised theories of translation that create commensurability between source and destination XML content and merged ontologies, is the system that should be selected among a number of competitors. We now apply this criterion to a comparison of the three transformation/ translation systems reviewed above.

5.1 The Commensurability Creation Load

We have coined the term commensurability creation load to describe the extent to which infrastructure is required to contribute to the means by which commensurability can be created between different ontologies. We are concerned here with the costs and complexity of establishing, administering and maintaining such infrastructure. In framing our discussions about such matters, we are cognizant that different system architectures have significant impact on the nature of required infrastructure. Therefore, we discuss the issue of commensurability creation load under the headings of XML to XML to XML (CGML and COAX) and XML to ontology to XML (OntoMerge) infrastructure systems.

XML-based interlanguage infrastructure

With the two XML to XML to XML transformation architectures discussed in this paper, the infrastructure requirements are similar to the extent that both require the creation, management and upgrading of interlanguage terms, including data dictionaries. In both cases, the origins of these terms are derived from declared XML standards. Therefore, one of the overarching infrastructure requirements of the XML-based transformation architectures are global agreements about protocols for the publishing of XML standards along the lines briefly outlined for example in a draft working paper by the interoperability working group of the Victorian Council of Social Services (VCOSS, 2008, p5).

Beyond the publishing of XML standards we think that in the construction of interlanguage terms including data dictionaries, the commensurability creation load of the CGML system architecture will prove to be significantly less than for the COAX system. We think this for two reasons. First, with CGML, all interlanguage terms are derived from published XML standards, with CGML's focus on

noun mappings. Therefore with the CGML approach, the transformation architecture is grounded in a familiar social language of declared standards. This is not completely the case with COAX, where the verb triples are designed to be machine readable and not necessarily easily read by people. Second, there will need to be less inter-language terms and data dictionary entries in CGML as compared with the number of relators and triples within the COAX system dictionary. This claim is supported by some of the early figures, comparing the sizes of the data dictionaries of the two systems, and also by some simple arguments.

In 2005, the COAX system was reported to have a dictionary of 27,000 terms of which 23,000 were relators. It also included around 500,000 triples with a dozen standards being mapped into the system (Rust, 2005, slide 13). The first reason for this difference in scale of approach between the COAX and CGML systems is apparent just from looking at Figure 3. Let us assume that this figure refers to one simple XML standard (and not multiple metadata elements from several different standards). If there are n circular objects in each of the pictures, then this requires that n items be defined in the CGML dictionary. By contrast, the COAX dictionary requires that up to $n(n-1)/2$ triples be defined for every relator which is used to join n objects in the ontology. If n equals 10 and there are five relators in an ontology, then 10 dictionary entries would need to be defined for CGML. But for COAX, there could be $10(10-1)/2$ or 45 triples for each of the five relators; altogether up to 225 triples that need to be mapped to the COAX system. As n increases linearly, the required number of COAX mappings explodes exponentially. Of course, not all objects will need relating to each other in triples, but the point that the number of COAX mappings necessary to create commensurability can explode compared to the number of CGML mappings remains a compelling one.

Ontology-based interlanguage infrastructure

The infrastructure requirements associated with the XML to ontology to XML transformation architectures such as OntoMerge system will need to be consistent with the design principles of the semantic web. In order that functional outcomes are achieved, such requirements include the creation and publishing of, and access to, libraries of both standard and merged ontologies.

Ontology mappings will need to be published on the semantic web just as ontologies themselves are (Bernstein and McDermott, 2004, p 1).

As we have also highlighted, the merged ontologies will need to reflect the underlying ontological structure of OntoMerge with its focus on semantic mappings of predicates and the formulation of both

bridging axioms and meaning definition axioms using these predicates.

We think that the commensurability creation load of CGML will be less than for OntoMerge. This is because with CGML, only the nouns are mapped from source and destination XML standards to the CGML interlanguage. In principle, once there are sufficient XML standards mapped into CGML, then possibilities associated with XML schema to XML schema translations could occur. In contrast, with OntoMerge, the mapping and merging of ontologies, involves the formulation both of bridging axioms between the source and destination standard ontologies as well as the MDAs between the surface ontologies and the standard ontologies. On the other hand, OntoMerge doesn't require the mapping of triples and the creation and management of a large data dictionary as required by COAX. Therefore the commensurability creation load in OntoMerge should be much less than we find in COAX.

5.2 System design to integrate the use of human interpretative intelligence

Different XML standards will all be 'richer' in their own distinctive ways because some will be slightly different and some will be very different from each other. Because of this, judgments about semantic and syntactic subtleties of meaning will be required in order to revise and reformulate the interlanguages in all three systems.

This means that human interpretive intelligence will, in principle, be necessary to enact the transformation of content from one storage system to another. So, a very important question for comparing transformation/translation architectures in general, and our three illustrative systems, in particular, is the provisions they make for incorporating the human interpretative intelligence needed for creating commensurability.

As we have seen, all three systems considered in this paper all incorporate human intelligence in the course of developing the various interlanguages. In the cases of CGML and COAX, human interpretative intelligence is incorporated when mapping rules are undertaken between the elements in different XML standards and the CGML or COAX interlanguages respectively. This principle could be extended beyond XML standards to XML schemas as well.

In the case of OntoMerge, human interpretative intelligence is central to the process of creating meaning definition axioms and bridging axioms.

Therefore, even though the three systems are alike in that they all incorporate human interpretative

intelligence in the creation of each transformation systems, we have already seen that they are not alike in the ways in which their respective transformation mechanisms work. In order to create a transformation architecture that is capable of handling the problem of incommensurability, there is a need to make a technical design choice about how to execute the semantic mappings and rules between the source and destination schemas. The ontological assumptions of the three different systems differ. The CGML system uses noun to noun mapping rules to relate the underlying digital elements of content. In contrast, the COAX system uses verbs and the linkages between digital elements that are generated when verbs are used (we have described these as ‘verb triples’). Finally, OntoMerge also uses nouns, but these act as noun-predicates and thus the noun to noun mapping rules in OntoMerge are fundamentally different from CGML.

We now need to consider the ease with which the use of human interpretative intelligence can be incorporated into the design and use of these different types of transformation mechanisms. We are specifically interested in what happens during the process of translating XML content.

5.3 Facilities for the use of human interpretative intelligence in content translations

COAX

The COAX system is not designed to facilitate human input. However, Rightscom (nearly) admitted the problems associated with aspiring to a fully automated translation system by highlighting that the COAX system *should* (but by implication does not) facilitate what we perceive to be a type of human intervention.

... there is a strong argument for establishing an (optional) human interface for the management of default assumptions, or the making of ad hoc choices in specific cases. In principle, it is bad practice for defaults to be applied “blindly”: at the very least the user should be aware of the assumptions that have been applied. Each user may have different preferences, in accordance with their own practices, and so the provision of “configurable” defaults and choices would support transparency and respects genuine variation in practice. (Rightscom, 2006, p. 31-32).

We agree with everything in this quotation, except the word ‘optional.’ Note that this word has been

placed in brackets. The reasons for this are not clear.

To take this argument further, consider the right hand side of Figure 3. If one circle is ‘dog’ and the other is ‘cat’ then a moment’s thought reveals that a number of triples could emerge (the line connecting the circles). The cat can ‘look at’ or indeed ‘spit at’, a dog. But will a cat “purr in the presence of” a dog? In some cases, this in fact might happen, but it will depend on context. We would expect difficulties like these to arise repeatedly.

We suggest therefore, that it will be a much more difficult task to describe all the triples in the COAX system than it will be to create data dictionary entries for the CGML system. We believe that this complexity is likely to be so great, that, in the final analysis, the COAX system will need to rely upon a mechanism whereby somebody defines allowable triples and then mandates their use in a standard.

Not surprisingly, something like this approach has already been proposed as part of the COAX system. The origins of this system in fact go back to the <indecs> metadata project. <indecs> was a project supported by the European Commission, and completed in the year 2000, that particularly focused on interoperability associated with the management of intellectual property.²⁶

It is this <indecs> metadata framework and the focus on rights and permissions associated with IP management that has given rise to the notion of ‘assertion’. “Assertions are the mechanisms in the <indecs> framework by which authority is established” (Rust and Bride, 2000, p, 35). Rust (2005, slide 13) makes this point clear: “all triples are ‘asserted’ by at least one authority” Such a process of ‘assertion’ becomes a means by which standards are established.

It might be no accident that the authors of the COAX system think that triples will need to be ‘asserted’. We think that defining triples will often be – not always, but often - difficult, and may require a great deal of human knowledge and judgment. Some human – indeed a whole team of humans - will, we think, probably need to manage the triple definition process. This essentially has become the role of the MPEG-21²⁷ Rights Data Dictionary Registration (RDD) Authority²⁸. MPEG-21 is an open framework for multimedia delivery and consumption, developed under the joint standards

²⁶ <indecs> was designed as a fast track, infrastructure project aimed at finding practical solutions to interoperability affecting all types of rights-holders in a network, e-commerce environment. The project focused on the practical interoperability of digital content identification systems and related rights metadata within multimedia e-commerce (Info 2000, p 1).

²⁷ MPEG standards for Motion Pictures Experts Group

²⁸ See <http://www.iso21000-6.net/>

framework of the International Standards Organisation (ISO) and International Electro-technical Commission (IEC) Joint Technical Committee. Two components of the MPEG-21 standards, a Rights Expression Language - REL - (ISO/IEC 21000-5) and Rights Data Dictionary (ISO/IEC 21000-6) provide the means for users to express rules about access and use of multimedia content. The REL provides the syntax, and the RDD provides the semantics for the Terms used in the REL. (ISO/IEC 21000-6 Registration Authority, 2007).²⁹

The World Intellectual Property Organisation (WIPO, 2003, p1) has highlighted the importance of a rights expression language in the following way.

A rights expression language requires extremely precise terms (semantics) in order to create precise, unambiguous expressions. However, it has long been recognized that natural language and computer language are two different things. The language of the everyday is far from precise in computer terms and society is built on the notion that the interpretation of the nuance of language is essential. For instance, all law is framed on the basis that it cannot be so precise as to exclude interpretation. Computers, on the other hand, cannot deal with imprecision. Given an ambiguous expression, computers will either fail to work or will function in an unpredictable manner. For this reason, it is necessary to create a set of terms (words) specifically for use in a rights expression language.

We conclude from this, that assertions result in the pre-specification of relationships between entities (as reflected in the lines in Figure 3). This pre-specification results in some user actions being allowable and others not – based on the notion of authority. This is the very basis of a licensing agreement itself. But we claim that the mechanism used to achieve this result (this use of verb triples) has an unintended consequence of imposing a constraining effect on the entire system. This flies in the face of one of the guiding principles of the internet itself: namely the principle of minimal constraint. Berners-Lee (1996) highlighted the principle of minimal constraint as follows.

... as few things should be specified as possible (minimal constraint) and those specifications which had to be made should be made independent (modularity and information hiding). The

²⁹ A methodology and structure for the (Rights Data) Dictionary is also standardised, along with the method by which further Terms can be added to the Dictionary through a Registration Authority. The methodology is based on a contextual ontology architecture, and was developed from earlier work on <indec> (ISO/IEC 21000-6 Registration Authority, 2007).

independence of specifications would allow parts of the design to be replaced while preserving the basic architecture.

Within this broad context, we believe, that however important rights management issues are, they should not be allowed to dominate the interoperability agenda. In particular, a consequence of this view is that it will prove highly problematic to create a global interoperability architecture that is fully automated; any transformation system – we believe - must build within it a provision to apply human interpretive intelligence, a point we'll expand upon below.

Rightscom has completed a project in which its transformation system has been piloted via real case studies. The published paper arising from this pilot project (Rightscom, 2006) frankly identifies a number of difficulties which the COAX system faces. The report is written in such a way that it appears that the authors are surprised by the difficulties they report. Indeed, they appear to believe that, once teething problems are sorted out, all these difficulties will be eliminated.

Here is our summary of Rightscom's findings.

Rightscom found challenges associated with *semantic incommensurability*.

Transformation can only preserve semantics: it cannot add semantic content which is not there to begin with. Where the semantics of two standards are incompatible with, or orthogonal to, one another, no transformation methodology can bridge the gap, and meaning will be "lost" in transformation because it has nowhere to go. (ibid, 2006, p.28)

The point here (not theirs, of course) is that, no automatic transformation methodology can bridge the gap because no wholly rules-based approach can formulate new rules. It takes "human interpretative intelligence," or human creativity, to come up with new solutions to transformation problems, or, more generally, as coming up with new theories of transformation that create commensurability.

Rightscom also finds challenges associated with *semantic ambiguity*.

Example 1: Dublin Core differentiates creator and contributor according to the importance of the role, whereas ONIX differentiates by the type of the creative role (for example, Author and EditedBy), so there is a semantic mismatch in mapping from ONIX to DC (though not vice versa). (ibid, p 31).

Example 2: To know whether an ONIX contributor is a Person or an Organization, it is necessary to look first at the PersonalName and OrganizationName fields to see which one is populated. In this particular example, Rightscom reports that conditional rules are required in some instances to determine the semantics of particular fields to ensure transformations are achieved. In such circumstances, relators are not resulting in an unambiguous expression of meaning; rather it is the conditional rules applied to determine the semantics of the particular fields. (ibid, p. 31).

Rightscom also highlights challenges which appears to us to be the same as *semantic delicacy* in the CGML system. The COAX system does not always handle the breaking down of different fields and managing the relationships between the hierarchies of these fields:

ONIX (like COA) separates out the quantity, the unit of measure and the type, so (for example) “Height 25 cms” goes in three fields in ONIX (and COA) but only one (or sometimes two) in MARC and DC. Clearly, in one direction this is fine: accurate outputs for MARC and DC can be generated by producing a concatenated field from ONIX data within COAX. From the other end it is less straightforward, but where consistent practice is employed in MARC, it is possible to analyse “Height 25 cms” into controlled value COAX fields. Comparable approaches can be taken for manipulating elements such as date/time variations, personal names and certain identifiers, with the likelihood of a high level of accuracy in results. (ibid, p. 30-31).

We agree that these difficulties, pointed out by Rightscom, are all real.

In the COAX system, triples contain the semantic and syntactic information which is required for content transformation³⁰. It is clear that the use of triples is necessary to automate content transformations in COAX. Indeed, it is clear that triples will, in many cases, embody the kinds of information which would be stored in the four filters in the CGML system. As but one example, Rightscom (2006, p 54-5) stated:

A COA ontology is built by creating hierarchies of specialized classes of Entity in the form of SubClasses or SubRelators to the Primary EntityTypes. For example:

Entity HasSubClass Resource

³⁰ It is clear that the use of triples was *designed* by the originators of the COAX system to store this kind of information, and so to automate content transformations as completely as possible.

Resource HasSubClass Output

Output HasSubClass Creation

establishes this hierarchy:

Entity

Resource

Output

Creation

This set of triples embodies the kinds of information contained within the super-ordination filter within the CGML system.

Superordination relations perform the function of sub-classification. They express an ‘is a’ relationship between one level in the taxonomic hierarchy and another. <Book> is a <Product>, as is an <AudioRecording>. (Cope and Kalantzis, 2004, p 251).

If the COAX system meets the ambitions of its designers, then it does not matter how easy or how difficult it is to apply human interpretative intelligence. This is because - by assuming success in achieving complete automation in translation – such requirements will not be necessary. It is our judgment – and this is an empirical judgment – that the COAX system will fail in this ambition. As a result, some form of human intervention, will be necessary to accomplish the continuous evolution of COAX. Further, currently, as we'll see below, COAX doesn't have the capability to receive and respond to the continuous criticism that would allow it to change to move closer to a true transformation system.

In the face of this probable need for applying human interpretative intelligence in COAX we now point to a fundamental difficulty in applying it in the COAX system. In the COAX system, the means by which semantic and syntactic interpretation occurs will be via a two staged process. First the user will need to choose an appropriate triple that contains the subtleties of the semantic and syntactic information to support transformation. And, second, the appropriate XSLT transformation mechanism will need to be applied to enact the transformation.

We think that such a mechanism is likely to result in user resistance for the same reason that we have discussed previously – the notion of intangible knowledge. When choosing an appropriate triple, users

will have to draw parallels between their accrued experiences of the domain in question and the triples that contain the semantic and syntactical subtleties associated with the users' interpretations of their experiences. But, triples used within the COAX system are not designed for human readability.

Rightscom itself has highlighted this point.

“The COAX XML schema is built on this model, which allows for any semantic relationship at any level of granularity to be represented. The result is much more verbose than a typical metadata record, but as the purpose is for storing data for interoperability, and not human readability, this is not a problem (Rightscom, 2006, p 26).

Since, by its account, COAX triples are not designed to be readable by people – but only by machines, we believe that this system will face serious user resistance, and also won't be able to continuously incorporate the human interpretative intelligence needed to revise, maintain, and re-create the merged ontology necessary for continuous successful translation of XML standards (and by implication, XML schemas as well).

OntoMerge

The OntoMerge system aims to maximize automation of ontology and content translations as much as possible. We choose the word *maximize* carefully because Dou et al, (2004a, p 2) emphasise the importance of human interpretative intelligence.

Devising and maintaining a merged ontology must involve the contribution from human experts, both domain experts and “knowledge engineers”. Once the merged ontology is obtained, ontology translation can proceed without further human intervention.

Equally, Qi et al (ibid) highlight that OntoEngine provides a mechanism for automated translations through its automated reasoning function.

We call our new approach *ontology translation by ontology merging and automated reasoning*. Our focus is on formal inference from facts expressed in one ontology to facts expressed in another (ibid, p 2).

The overall objective of this approach is to provide automated solutions in the translation of datasets and queries.

If all ontologies, datasets and queries can be expressed in terms of the same internal representation, semantic translation can be implemented by automatic reasoning (Dou et al, 2004a, p 16).

This automated reasoning function of the OntoMerge system architecture makes it fundamentally different from CGML and COAX. At base level, it is the reliance on bridging and meaning definition axioms, as predicate to predicate mappings, that forms the basis of the functionality of OntoEngine. This is because predicates are an important part of the OntoEngine indexing structure that allows it to find formulas to be used in inference (Dou et al, 2004b p 8).

Given the automated reasoning functions of OntoEngine, and the heavy reliance on these, we think there is little provision to integrate the use of human interpretative intelligence during the content translation querying process. However, we also conclude that there would appear to be little perceived benefit in this for the developers of OntoEngine because it would be contrary to the ambitions of the developers of OntoMerge. We have previously highlighted their overall objective of processing the content of information and maximizing possibilities associated with automation.

But this is not to suggest that OntoMerge is an example of a system/architecture that does not result in the falsification of the underlying translation theory that makes up the architecture. In contrast, the OntoMerge system relies on the continuous merging of new ontologies and datasets to create a broader ontological framework. The ontology merging process in principle provides a means of falsifying and continuously expanding the underlying ontology and results in new solutions to the content translation problem as outlined in Section 2.2 and 2.3.

A question therefore arises from this. How easy is it for human interpretative intelligence to be incorporated into attempts made to automate the formulation of meaning definition and bridging axioms upon which the OntoMerge system relies? Dou et al (2004b, p 17) address this question in the following way.

We designed a semi-automatic tool which can help generate the bridging axioms to merge ontologies. It provides a natural-language interface for domain experts who are usually not good at logic formalism to construct and edit the axioms.

The reliance on a natural language interface to support the integration of human interpretative intelligence into the OntoMerge system exposes two key differences of approach as compared to

CGML. First, the creation of bridging (and meaning definition) axioms requires that the domain experts understand that they are mapping noun-predicates to noun-predicates as compared to nouns to nouns. Second, the underlying Web-PPDL language used by OntoMerge is difficult to understand – thus the primary reason for developing a natural language interface.

As a result, the integration of human interpretative intelligence into the OntoMerge architecture will result in the need for domain experts to understand something about the principles of semantic web ontologies. We think this problem will become compounded by the need to generate a distributed library of semantic web and merged ontologies to support OntoMerge. We note that in contrast, XML standards are already being negotiated, published and used by industry practitioners – even though more needs to be done to provide protocols for how such standards should be published.

Facilities for Human Interpretative Intelligence in CGML

The difficulties in the use of human interpretative intelligence found in COAX and to a lesser extent in OntoMerge are largely dealt with in the CGML system. All CGML terms are defined drawing upon the social languages of practitioners – because the CGML interlanguage terms are defined on the basis of their relatedness to digital elements of XML standards. The CGML filters - delicacy, synonymy, contiguity, and subset – exist to expose areas of incommensurability that arise during XML content translations. When the filters expose areas of incommensurability, users are prompted via structured user queries to make choices and to take action. In other words they apply human interpretative intelligence during the content translation process.

With the application of such actions slight distinctions of semantic and syntax differences between the tags within each XML standards become codified. If the “digital tracks” of each transformation are archived, we have noted in the CGML system, that the ‘filters’ can be ‘trained’ to respond appropriately to these differences. In this way, there will be a movement towards more successful automation, because human creativity is constantly being used to revise and improve the merged ontology underlying the translation system. However, such automation can only develop from the accumulation of digital tracks that develop from actions arising from human interpretation and choice. Over time, well-trained filters could, in principle, come much closer to automating the content transformation process.

It is our judgment that automation will never be complete, however. Some human intervention in the

form of semantic and syntactical interpretation will always be necessary. This follows from the idea that interlanguage is a theory about translation, and therefore is fallible and sooner or later will encounter a translation problem that is likely to falsify it. But since mediated actions revising such theories are part of the design of the CGML system itself, such design features allow (a) many interpretative choices to be taken by users rather than programmers, and (b) for the continuous revision of the transformation/translation theory that is at the heart of CGML.

We think that the need to apply human interpretative intelligence is also supported by another very important aspect of developing and revising theories of translation. This relates to the application of the user's intangible knowledge. In making choices associated with the subtleties of the semantic and syntactic distinctions when content transformation takes place via human intervention, users must and will draw upon their own experiences and subject matter expertise associated with the domain of practice (i.e. the industry that the XML standard relates to). In this way, the personal knowledge of users themselves is available to support the "intelligence" of the CGML transformation system, and to incorporate the capability of the users to revise, extend and re-create merged ontologies.

Conclusions on Facilities for using Human Interpretative Intelligence

Of the three illustrative systems, CGML provides a facility for using human interpretative intelligence, in principle, both in the ontology merging and the content translation processes. These processes can both be supported through the use of a friendly interface so that users can be queried in order to resolve semantic ambiguities that arise in constructing transformation rules, or executing translations.

In COAX the content translation is fully automated, but human interpretative intelligence is used in creating mapping rules using triples. Up until now, it appears that the process of creating these rules involves the need to be able to understand the meaning of triples. However, COAX triples are not easily read and understood by humans. This effectively constrains those involved in the COAX mapping rules to technical personnel.

In OntoMerge, like COAX, content translation is fully automated and provides no opportunity for human intervention. However, formulating bridging axioms and MDAs, employs human interpretative intelligence in the form of domain experts whose participation is facilitated by an axiom generating tool. This tool provides a natural language interface that ensures users do not have to learn the underlying Web-PPDL software language.

In conclusion, among the three systems we have used in this paper for illustration and analytical purposes, CGML has the best facilities for incorporating human interpretative intelligence. OntoMerge's facilities don't allow participation in the translation process, but are effective in incorporating the knowledge of domain experts in creating bridging axioms and MDAs. Finally, COAX is the system whose design is most oriented to the ideal of full automation, and, as such, it is the system that limits human interpretative intelligence to the work of programmers in constructing mapping rules.

6 ADDRESSING THE TRANSLATION PROBLEM: EMERGENT POSSIBILITIES

We think that the evolution of information architectures required to generate solutions to the problem of translatability or interoperability of different XML languages, as outlined in this paper, depends on how different approaches deal with the need to apply human interpretive intelligence where it is needed. In our view, a number of different outcomes might emerge; we think it will prove impossible to predict what architectures are likely to prevail. However, we discuss some of the possibilities in turn below.

6.1 Possibility One. No choice - A patch and mend architecture prevails

A patch and mend (or rule based tag-to-tag) outcome is likely to evolve if no systematic effort is made to address the key challenges to achieving translation of HU content addressed in this paper. This outcome will depend on the kinds of unsystematic and localised approaches to interoperability we have discussed in Sections 2.1 and 2.3. But, as we have previously suggested such localised solutions risk producing outcomes in which the infrastructure required to support the exchange of content within a global economy remains a hopeless jumble of disconnected fixes. The reason for exploring more systematic approaches is a pragmatic one. The costs of re-configuring content to a sustainable data structure are substantial. A global architecture to support translation and interoperability objectives would help to address the challenge of “content sustainability” – and to avoid the cost of re-configurations required to keep content accessible and usable.

6.2 Possibility Two. Choice to maximize the automation of translations

A choice to pursue the use of system architectures that aim to minimize the need for human intervention and thereby maximize the potential for automation is likely to prove attractive - for

obvious reasons. Within this choice, we see three potential scenarios unfolding.

Scenario 1: Successful solving of the translation and interoperability challenge

A system architecture that successfully solves the translation and interoperability problem will emerge if such problems can be addressed for one, or for a number of particular industries. This would give such a system(s) a critical mass from which further advances could be built. This would especially be the case if this type of functionality would suit a certain group of stakeholders. In the case of COAX, for example, those players that have control over aggregated quantities of copyrighted content in the digital media industries, such as publishing and motion pictures would likely benefit from the successful implementation of COAX. This solution could also benefit a wide range of end users of content as well - on the basis that access to such content could be delivered on an on-line basis. End users such as University students might be able to access customised content such as book chapters on an on demand basis. Or current consumers of motion picture products would benefit from increased on-line access to copyright content – thereby by-passing current distribution retail channels. Owners of copyright content could leverage and protect the value of their content. Everyone would win.

But, for reasons outlined in Section 5.3, we believe that the introduction of such solutions is likely to be met with resistance by the users of the systems, because of the constraining impact on the system. We suggest that as a result of this two different scenarios could unfold.

Scenario 2: in the end, the systems do not work

It is possible that a significant amount of resources are allocated to the development and roll out of system architectures of the COAX or OntoMerge type. OntoMerge, for example, forms part of the network structure of the semantic web initiative (an initiative that has already consumed vast resources). But, the more resources are expended, it might well be that there is a continuing need to justify the expenditure of such resources, even, though, in the end the systems might not work. An historical example of this type of waste already exists – namely the EDIFACT story.

In the mid 1980's work was undertaken in a UN-sponsored initiative called EDIFACT (David and Forey, 1994). The objective of EDIFACT was to create a global standard to support computer-to-computer exchange of business documents. Now, twenty years later, these aspirations have not been achieved and a paperless trading system, operating by means of computer-to-computer exchanges - which EDIFACT was meant to help set up - has not been realised. EDIFACT is widely regarded as

having been a failure.³¹

What emerged during the period in which EDIFACT was championed was the rise of a plethora of different, local, standards. Traders used file transfer protocols (FTPs) for specialized documents transmitted, for example between marine insurers and customs brokers and their clients. Initially, this was conceived as an overlay on the telex and telephone communications messages, but such transfers eventually replaced the use of these technologies. The document formats substituted for and enriched the key bits of data that were conveyed using previous technologies. The striking of deals, and the enforcing of them, would have been much more difficult without such localised standards with which to specify what information was needed to support a deal, and in what way.

EDIFACT aspired to create much more of a system than this. But it turned out to be impossible for EDIFACT to play a role in supporting business exchanges, for two reasons which are of relevance to this paper.

(i) As already noted, standards which enable computer-to-computer exchange of important business-information need to be developed in ways that reflect highly localised requirements of specialists. EDIFACT tried to create a universal document that could describe all the possible variations of transactions. But it turned out that the EDIFACT system took too long to develop and became too cumbersome. As a result such efforts were evaded by the use of highly localised standards, which reduced the costs of transactions that were already occurring without support of electronic networks, but which did not make use of proposed universal approaches.

(ii) Computer-to-computer interoperability of content and information usually becomes entangled with exchanges between the actual human beings who are operating the computers themselves. EDIFACT was conceived in ways which did not make it sufficiently flexible for this to be possible.

David and Foray (1994, p 138) summarise the implications of this failure.

According to our assessment of the European situation, the main mission of EDIFACT should not be that of displacing the local standards, but rather one of assisting their absorption within a unified framework. Increasing the flexibility of the language and tolerating a sacrifice of aspirations to rigorous universality may therefore prove to be the most effective long-run

³¹We are grateful to Paul David for helpful insights associated with the material in this section.

strategy. However, due to the persisting ambiguity that surrounds the goals of those charged with developing the standard, EDIFACT policy continues to oscillate between conflicting design criteria. This situation, in and of itself, should be seen to be working against the formation of greater momentum for EDIFACT's adoption and the diffusion of EDI.

We see a very real possibility that an attempt to create a global translation and interoperability agenda, based on either the system like the COAX or the OntoMerge type system, might end up with an outcome of this kind.

Scenario 3: On-going resource allocations are made to gain a minimum level of functionality

If the resistance of users was not sufficient to cause the system to become dysfunctional, an alternative outcome might develop. This could involve the continuous investment of resources in system architecture to gain a minimum level of functionality, to the extent that this prevented other, more suitable, architectures from being established. The result might be a lock-in to an inappropriate system. This is like the outcome in video recording software, where the poorer tape-recording system prevailed (i.e. the VHS tape-system rather than the BETAMAX tape-system). There is a famous historical example of this lock-in story – namely the story of the QWERTY keyboard. This keyboard was designed for good reasons – very common letters were separated on the keyboard, in order to prevent the jammed keys which were likely to arise, in a manual keyboard, if keys next to each other were regularly pressed one after the other. In 1935, The Dvorak Simplified Keyboard -DSK – became available and was a superior standard for keyboard design – leading to potential significant productivity improvements.

David (1986, p14) summarises what can happen in these cases as follows.

Despite the presence of the sort of externalities that standard static analysis tells us would interfere with the achievement of the socially optimal degree of system compatibility, competition in the absence of perfect futures markets drove the industry prematurely into de facto standardization on the wrong system - and that is where decentralized decision-making subsequently has sufficed to hold it. Outcomes of this kind are not so exotic.

In the cases discussed in this paper, where there is already a COAX or OntoMerge type architecture available, we argue that a similar story to the QWERTY one is a possible outcome. Suppose that a CGML type system has the potential to provide a better overall architecture. But such a system might

not emerge unless enough users see the benefit of adopting it. However, this will not happen if translation or interoperability needs are already being met, more or less, by on-going resource allocations to achieve minimum functionality requirements of competing type systems.

It is interesting to note that both the COAX and OntoMerge systems are structured in ways that, in principle, form part of an approach to global interoperability infrastructure embedded within the semantic web vision. The semantic web vision is a significant one and already there are powerful networking effects arising from this global agenda. These network effects might provide a powerful rationale to persuade agents to adopt this approach. And they might do this, even if other approaches like the CGML system were superior. This type of constraining effects on standards setting has been described by David (2007, p 15). His conclusion as to the policy implications is as follows (ibid, p 20):

If there is one generic course that public policy should pursue in such situations, it would be to counter-act the “excess momentum” of bandwagon movements in network product and service markets that can prematurely commit the future inextricably to a particular technical standard, before enough information has been obtained about the likely technological or organisational and legal implications of an early, precedent setting decision.

We see such an outcome as possible, in moves towards the creation of global interoperability architecture. We think it desirable that policy makers should be alert to this possibility.

6.3 Possibility three. Choice to embed human interpretative intelligence

A third possibility that could evolve is a system which systematically embeds the principle of human interpretive intelligence, such as the CGML system described in this paper. The CGML transformation system architecture (for example) has the potential to evolve and develop because of its flexibility. This flexibility is derived from the fact that the system specifically contains within it a provision to facilitate the application of human interpretative intelligence during the content translation process via structured queries. We have argued that the ability to do this easily is linked to the technical design choice of focusing on nouns (in the case of CGML) versus verbs (in the case of COAX) and nouns as predicates (in the case of OntoMerge). User responses to structured queries become part of the accumulated “recordings” of semantic translations. These responses might be built up into a ‘bank’ of previous translations. The results stored in this bank might then be used to refine the operation of the filters, which we have discussed previously, so that, as the transformation process is repeated, fewer and fewer

structured queries are thrown up for the user to respond to. Thus, the more the system is used, the more automated the CGML system might become. This transformation system is premised on the principle of minimal constraint (refer to Section 5.3) precisely because it aims to facilitate human intervention where necessary, but at the same time to do this in a way which might limit such intervention. And the flexibility of the design of the system is such that it also has the potential to be used as a universal application across multiple industries. This is because it would enable the absorption of the kinds of localised schemas within a unified framework.

Despite these positive suggestions, there is a significant challenge associated with the CGML system itself – or any other system that systematically takes into account the principle of human interpretative intelligence. That is, at the current time, there are limited network externality effects available that would result in a widespread adoption of such an approach. Therefore, if such an approach to the problem of XML translation is to become widely adopted, then the underlying theory of translation, must not only withstand critical evaluation - it must also become a practical solution to the challenges of XML language translation and interoperability at both localized and global needs. We do not have any immediate answers to this question. However, in a separate paper we aim to address some suggestions about how to progress practical and pragmatic pathways for establishing a foundation upon which open approaches to interoperability might progress. We also note with interest attempts in Finland (Hyvönen, et al. 2007) to develop a national semantic web content infrastructure and that the elements of such infrastructure include *shared and open metadata schemas, core ontologies and public ontology services* (ibid).

7 CONCLUSION

This paper has discussed how to achieve interoperability in the transformation of XML-based digital content from one XML language to another. It is our view that it will prove impossible to create a transformation system that is fully automated. Any transformation system – we believe - must build within it a provision to apply human interpretative intelligence.

We have also highlighted that it will also be necessary to make a technical design choice about the means by which the translation process between different XML languages is achieved. In making such a claim, we have examined three transformation systems that have three entirely different ontological approaches to address the same translation problem: the Common Ground Markup Language, (the

CGML system), Contextual Ontology_X Architecture (the COAX system) and OntoMerge system.

The important feature of the CGML system is that it specifically contains within it a provision to facilitate the use of human interpretative intelligence during the translation process - by using structured queries and action response mechanisms. By describing in detail how this system works, we have identified that the categories used in CGML make use of published XML standards used by the relevant user communities, and have shown that elements of the CGML interlanguage apparatus are derived from these standards. This is important because different users generate different ways of organising content which reflect localised and specific activities. We have argued that the translation of the elements of content from one set of categories into another will not, in general, be possible without the application of human interpretative intelligence. We provide a detailed explanation of how this can be achieved in the CGML system. We have shown how this system ensures that users can draw upon their own particular knowledge. In this way, choices associated with the subtleties of the semantic and syntactic distinctions, which happen when content transformation takes place via human intervention, can enable users to draw upon their own experiences. We have claimed that any system designed to facilitate compatibility of content description and formatting frameworks must not dispense with the application of this type of human intelligence.

In contrast, we have shown how the COAX system relies on the use of triples to contain the semantic and syntactic information required for content transformation. But, because we conclude that human intervention is required, we have considered how difficult this process will be to implement within the COAX system. We conclude that because triples are used and because triples are not designed to be readable by people – only by machines – the COAX system rules will not easily be applied to support mechanisms for the application of human interpretative intelligence. As such, we argue that the system will become more constraining. This we argue is likely to result in user resistance.

We have also highlighted that in the OntoMerge system, a situation exists somewhere between the CGML and COAX systems. That is, like COAX, content translation aims to be fully automatic and provides no opportunity for human intervention. *However*, in constructing bridging axioms and MDAs that are required for automated content translations, human interpretative intelligence is employed in the form of domain experts whose participation is facilitated by an axiom generating tool that provides a natural language interface for use by human domain experts.

We have suggested that the primary criterion for comparing alternative translation/transformation

systems for XML standards and schemas is the ease of formulating expanded frameworks/ revised theories of translation that create commensurability between source and destination XML content and the merged ontologies. We have, in turn broken this criterion down into two sub-criteria: 1) the relative commensurability creation load in using each system, and (2) relative system provisions for and constraints on the use of human interpretative intelligence. On both of these sub-criteria, the CGML system is better than the other two, OntoMerge is next, and COAX ranks as the system with the highest commensurability creation load and the least relative provision for incorporating human interpretative intelligence.

We believe that the two claims of this paper - about the need for human intervention and about the need to make a technical design choice about the way in which content is translated - have important implications for the development of any global interoperability architecture.

The means by which we have developed our argument – by comparing three systems should not be understood as advancing the interests of one over another. We do note that both the COAX and OntoMerge systems are conceived in ways that are congruent with the design features of the semantic web initiative. We suggest this might give these approaches certain advantages arising from the externality effects that arise as a result of this. But in advancing our central claims, we wish to caution against premature standardisation on any particular system, before all potential systems have been properly compared. Such standardisation might lead to premature lock-in to inappropriate global interoperability architecture.

Our comparison of the three systems has suggested that there are essentially three outcomes possible depending upon the choices made. The first outcome is that no choice is made at all about the inter-related question of human intervention and ontology. We have suggested that the actions that arise from such a choice will lead to a patch and mend architecture - an architecture that will develop from unsystematic and localised approaches to interoperability. But, we believe that searches for more systematic approaches will prevail for pragmatic reasons – primarily because the costs of reconfiguring content in all sorts of different ways will eventually prove to be too great. We therefore suggest that a global architecture to support interoperability objectives is likely to become increasingly desirable. This is partly because of the need to address the challenge of “content sustainability” – and avoid the cost of re-configurations required to keep content accessible and usable.

The second outcome is a choice for working towards the benefits of automated translation systems, as

much as possible. In the case of the COAX system this draws upon the use of triples and in the case of OntoMerge, it draws upon the use of predicates (which might in their own right be defined as nouns). With this second choice, we suggest three scenarios could develop. The first is that the use of triples and or predicates could result in a workable system, with widespread benefits. But we have highlighted why both the COAX and OntoMerge solutions are likely to be constraining and why they might be met by resistance from users. A second scenario is that many resources could be allocated to the development and roll out of the COAX and OntoMerge type architectures, but that such resources could be wasted – primarily because, in the end, the systems prove to be dysfunctional. We have discussed a historical example of this happening – the EDIFACT story. The third scenario is that the resistance of users might not be sufficient to cause the system to become dysfunctional. This combined with the network externality effects of the COAX and OntoMerge systems being part of the International DOI Foundation and semantic web infrastructure might result in agents being persuaded to adopt a less than effective system, even if it is less desirable than a CGML-like system.

The third outcome is for something like the CGML system architecture to emerge. We have shown that this architecture has potential – because it offers a generalised solution which, at the same time, would allow particularised solutions to emerge, giving it some flexibility. We have highlighted how this flexibility is derived from the fact that the CGML system specifically contains within it a provision to facilitate human interpretative intelligence during the content translation process via structured queries and action responses.

Finally we note that our findings should not be interpreted as a “product endorsement” of the CGML system. The CGML system, like the COAX and OntoMerge systems, is not, in its current form, an open standard. This is problematic, because if such a transformation system is to offer a solution to support global interoperability objectives, the CGML system will at the very least need to be positioned as a type of standard in ways similar to those in which the COAX system has become part of the International Standards Organisations umbrella framework.

Our aim in this paper has been to suggest that there is a need for further thought before systems like the COAX and OntoMerge systems are adopted as the basis for any global interpretability architecture. Paul David has drawn out the lessons to be learned about the risks of premature lock-in to inappropriate standards. He suggests that:

preserving open options for a longer period than impatient market agenda would wish is a

major part of such general wisdom that history has to offer public policy makers. (David, 2007, p. 20).

We suggest that here too it would be desirable to preserve an open option for some time to come.

8. REFERENCES

- Bergon, R., 2001. XML.com. Other lessons learned from working with XSLT. <http://tinyurl.com/35dw69>. Viewed 5th June 2006.
- Berners-Lee, T. 1996. The World Wide Web: Past, Present and Future. World Wide Web Consortium. <http://www.w3.org/People/Berners-Lee/1996/ppf.html>. Viewed on 18th May, 2007.
- Bernstein, M. and McDermott, D., 2004. Ontology Translation for Interoperability Among Semantic Web Services AI Magazine 26(1): 71-82, Spring 2005. <http://tinyurl.com/6ompvh>
- Common Ground Publishing, 2005. Common Ground Markup Language. <http://commongroundgroup.com/ui/CGMLBrochure.pdf>
- Cope, B. and Kalantzis, M. 2004. "Text-made Text" *E-Learning*, Vol. 1, No 2. <http://tinyurl.com/2uh6ev>
- David, P., 1986. Understanding the Economics of QWERTY, or Is History Necessary? In *Economic History and the Modern Economist*, ed. W.N. Parker, Oxford: Basil Blackwell Publishers
- David, P. and Foray, D., 1994. "Percolation structures, Markov random fields and the economics of EDI standards diffusion", in Gerard Pogorel, ed., *Global Telecommunications Strategies and Technological Changes*, Amsterdam: North-Holland, 1994.
- Dou, D., McDermott, and Qui, P. 2004a. Ontology Translation on the Semantic Web. www.cs.yale.edu/~dvm/daml/ontomerge_odbase.pdf
- Dou, D., McDermott, and Qui, P. 2004b. Ontology Translation by Ontology Merging and Automated reasoning. <http://cs-www.cs.yale.edu/homes/ddj/papers/DouEtal-MAS.pdf>
- Firestone, J. and McElroy, M. 2003. *Key Issues in the New Knowledge Management*, Burlington, MA: KMCI Press/Butterworth Heinemann
- Hyvönen, E. et al. 2007. Proceedings of the First International Semantic Computing Conference (IEEE ICSC 2007), Irvine, California, September, 2007. IEEE Press
- Indecs, 2000. Interoperability of data in e-commerce systems. <http://www.indecs.org/>
- ISO/IEC 21000-6 Registration Authority, 2007. <http://www.iso21000-6.net/> viewed on 23 May, 2007.
- Info 2000. <indecs>. Putting Metadata to Rights. Summary Final Report. <http://www.indecs.org/pdf/SummaryReport.pdf>
- Kuhn, T. 1970. *The Structure of Scientific Revolutions*, 2nd Edition, Chicago, IL: University of Chicago Press
- MARC, 2006. Marc Standards: Library of Congress – Network Development and Marc Standards Office. <http://www.loc.gov/marc/>
- McLean, N. & Lynch, C., May 2004. Interoperability between Library Information Services and Learning Environments – Bridging the Gaps: A Joint White Paper on Behalf of the IMS Global Learning Consortium and the Coalition for Networked Information. http://www.imsglobal.org/digitalrepositories/CNIandIMS_2004.pdf
- ONIX, 2006. Onix for Books. <http://www.editeur.org/onix.html>
- Paskin, N., April 2006. Identifier Interoperability: A Report on Two recent ISO Activities, D-Lib Magazine. <http://www.dlib.org/dlib/april06/paskin/04paskin.html>

- Popper, K. 1970. In I. Lakatos and A. Musgrave (eds.), "Normal Science and Its Dangers," Criticism and the Growth of Knowledge, Cambridge, UK: Cambridge University Press
- Popper, K. 1972. Objective Knowledge, London, UK: Oxford University Press
- Qui, P., McDermott, D. and Dou, D. 2004. Assigning Semantic Meanings to XML. <http://cs-www.cs.yale.edu/homes/dvm/papers/xmlMeaning.pdf>
- Rightscom, April 2006. Testbed for Interoperability of ebook Metadata: Final Report. www.jisc.ac.uk/uploaded_documents/TIMEfinalreport24apr2006.pdf
- Rust, G. and Bride, M. 2000. The <indecs> metadata framework: Principles, model and data dictionary. <http://www.indecs.org/pdf/framework.pdf>
- Rust, G. 2005. Thoughts From a Different Planet. Presentation to the FRBR³² Workshop, Ohio. <http://tinyurl.com/2tsa9m>
- Wikipedia, 2006. XSL Transformations. <http://en.wikipedia.org/wiki/XSLT>. Viewed on 23rd Nov., 2006.
- Wikipedia, (2007). Semantic Web. http://en.wikipedia.org/wiki/Semantic_Web. Viewed on 15th Jan 2008.
- Victorian Council of Social Services, 2008. The Interoperability Challenge. A draft discussion paper by the Interoperability Working Group. <http://www.doingitbetter.net.au/docs/iwg-positionpaper.pdf>
- W3 Schools, (date unknown). Introduction to DTD. http://www.w3schools.com/DTD/dtd_intro.asp
- World Intellectual Property Organisation, 2005. Method and apparatus for the creation, location and formatting of digital content. <http://tinyurl.com/6emddp>
- World Intellectual Property Organisation, 2003. Current developments in the field of Digital Rights Management, Standing Committee on Copyright and Related Rights. http://www.wipo.int/documents/en/meetings/2003/sccr/pdf/sccr_10_2.pdf
- World Wide Web Consortium, (2007). Semantic Web. <http://www.w3.org/2001/sw/>. Viewed on 15th Jan 2008.
- World Wide Web Consortium, (2004a) OWL Web Ontology Language Overview. <http://www.w3.org/TR/owl-features>.
- World Wide Web Consortium, (2004b) OWL Web Ontology Language Guide. <http://www.w3.org/TR/owl-guide/>
- World Wide Web Consortium, (2004b) Resource Description Framework. www.w3.org/RDF.
- World Wide Web Consortium, (2004c) Resource Description Framework: Concepts and Abstract Syntax. <http://www.w3.org/TR/rdf-concepts>. Viewed on 15th Sept, 2008
- World Wide Web Consortium, (2001). DAML+OIL (March 2001) Reference Description. <http://www.w3.org/TR/daml+oil-reference> Viewed on August 13, 2008.
- World Wide Web Consortium, 2001. XML Schema. <http://www.w3.org/XML/Schema>. Viewed on 15th Jan. 2008
- World Wide Web Consortium, (Undated) W3C Semantic Web Activity. <http://www.w3.org/2001/sw/> Viewed on 14th Sept, 2008

³² FRBR stands for Functional Requirements for Bibliographic Records