OpenSEA: A framework for semantic interoperation between enterprises

BRIDGES, S, SCHIFFEL, J and POLOVINA, Simon <http://orcid.org/0000-0003-2961-6207>

OpenSEA: A Framework for Semantic Interoperation Between Enterprises

Shaun Bridges Open-SEA.org e-mail: Shaun.Bridges@Open-SEA.org

Jeffrey Schiffel The Boeing Company – Wichita Division e-mail: jeffrey.a.schiffel@boeing.com

Simon Polovina Sheffield Hallam University email: S.Polovina@shu.ac.uk

Abstract

The modus-operandi for information systems is shifting. Agility and adaptability will be the kingmakers in the decentralising enterprise architecture where on-premise and cloud systems have to be combined seamlessly. At the same time the wealth of data available to organisations needs to be understood and interpreted so as to provide information and inferences needed to generate the knowledge that drives competitive advantage. This chapter offers a high-level introduction to OpenSEA, a framework that combines the open semantics of TOGAF with the open syntax of ISO 24707:2007 Common Logic to provide an Open Semantic Enterprise Architecture. Because of its open nature it is free to adopt and extend, yet retains a root commonality to ensure all participating agents can agree on a common understanding without ambiguity, regardless of the underlying ontology or logic system used.

Introduction

A new frontier in enterprise architecture is being explored. This new frontier is the realisation of a distributed enterprise architecture. Like all frontiers it brings the possibilities of explosive growth and exploitation but also brings the undeniable fact that for every winner there will be a number of losers. Frontier dynamics were described by Pascale (2000), who likens organisations to organisms that need to adapt to survive or thrive. Pascale stated that evolutionary pressures are high where the surrounding environment shifts or opportunities to grow are offered in new environments. “A fish takes for granted the water in which it swims; when it learns about the land it is usually too late” (Pascale 2000, p.25).

So what can enterprises do to take advantage of this decentralised model? Organisations need to be able to respond rapidly to new offerings and to find and consume data, processes and services from other organisations. Such an undertaking requires protocols and common understanding. A number of new ontologies and definitions are being created to bring common semantics to the cloud architecture including a unified ontology of cloud computing (Youseff et. al., 2008) and a whitepaper on the taxonomy of cloud computing from the Cloud Computing Use-Case Discussion Group (2010).

It could be argued that this new architecture builds on Service Oriented Architecture since the web services that provided the foundation for SOA are now complemented by the XaaS services providing Infrastructure as a Service (e.g., Amazon Web Services), Platform as a Service (e.g., Force.com) and Software as a Service (NetSuite, SalesForce, Business By Design). The International Research Forum of 2008 explored the issues facing enterprises as they look to exploit this newly evolved Service Oriented Architecture and identified that service discovery is inefficient and that web services are too granular to be of value and need to be extended to provide functionality. They also described an ‘integration debt,’ where services are created in their own domains and data models. “The stack must rest on a firm architectural foundation and share a common language” (p. 57).

At the same time enterprises are reliant on data from within their own boundaries and data from the larger market place. Consuming and interpreting this wealth of data, i.e., drawing information from it, is central to an enterprises operation. Dashboards offer CEOs snapshots of every conceivable metric. The time relevancy of the data has created In Memory Databases, such as that used by SAP’s High Performance Analytics Appliance (HANA) to provide up to the second accuracy in data. But without a reliable understanding of what the data means it is of little value. Beyond information and data, knowledge is the driving force that creates competitive advantage. Taking data, extracting its meaning, and then using rules and inferences to derive new knowledge will allow an enterprise to predict where it needs to be rather than responding to the current situation.

Boisot and Canals (2007, p. 39) described knowledge agents as being able to use models and information to act on the prevailing environment. If an enterprise can be confident of the models that drive its knowledge agents, and of the value of the information fed to the agent, then it can be reasonably confident that it is acting in the best way to exploit its environment and explore new opportunities.

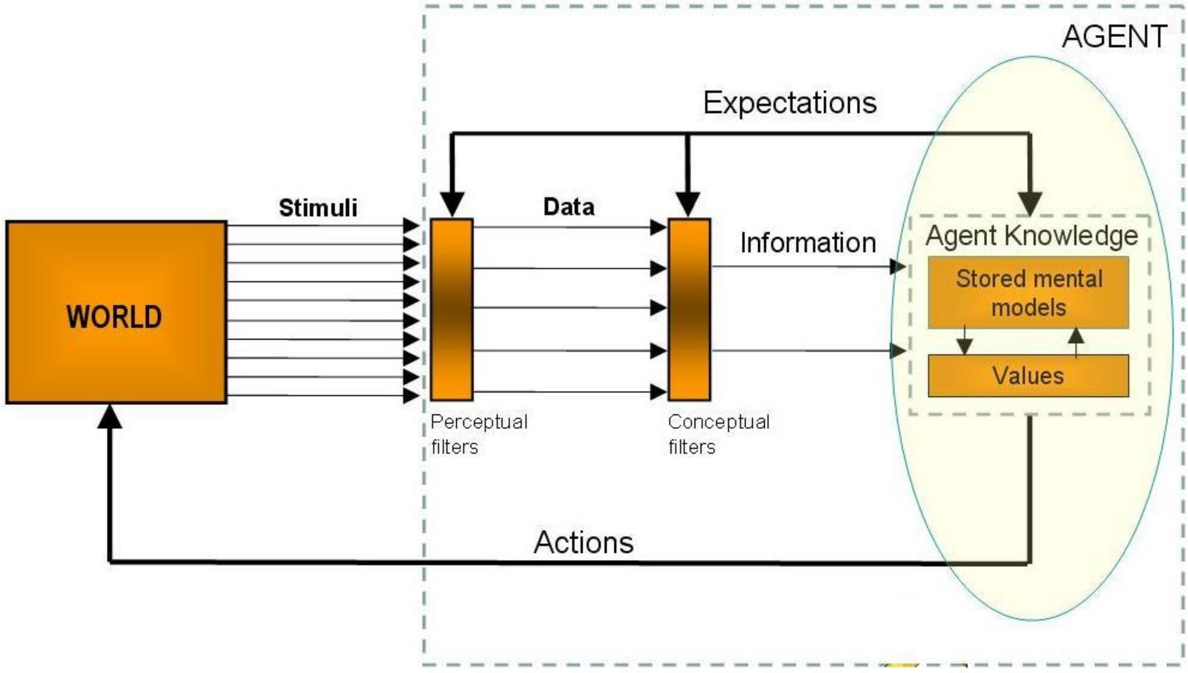

Fig. 1. Boisot's and Canal's Knowledge Agent; Actioning Filtered Stimuli to Effect the Environment.

Logic systems must be able to accurately filter out data from noise and combine these units of datum to draw the information. To achieve this the data may need to be annotated in such a way that the system can understand what the data 'means,' or information needs to be traded between different knowledge agents in a format that be used by both agents. Unfortunately, the wealth of different logic systems, and the lack of integration between them, means that different systems have to operate in isolation. As Sowa (2009) put it, "The proliferation of incompatible semantic systems is a scandal that is strangling the growth of the entire field" (p.119).

The International Research Forum also identified the need for semantic systems to be part of the future of Service Oriented Architecture including actions such as moving service discovery away from key words and domain specific ontologies, semantically enriching services (going beyond web services to include the XaaS offerings) and describing services in a holistic manner. This move towards adding meaning to data and combining different applications is a significant step towards Web 3.0. Google CEO Schmidt described Web 3.0 as "applications that are pieced together" (MacManus, 2009). Cap Gemini CTO Mulholland referred to the Web 3.0 Conference on his blog of July 2009 (Mulholland, 2009), noting the apparent shift in emphasis from machines to users as consumers of Web 3.0:

> The goal of Web 3.0 is to reorganize information so users can capture what things are and how they are related. This seemingly simple concept will have a profound effect at every level of information consumption, from the individual end user to the enterprise. Web 3.0 technologies make the organization of information radically more fluid and allow for new types of analysis based on things like text semantics, machine learning, and what we call serendipity — the stumbling upon insights based on just having better organized and connected information.

In summary, if data is to be annotated in a meaningful way it should be expressed in a format that is equally consumable by machines and humans alike. The links and relationships should allow discrete snippets of data or sources of information (including those from unstructured data such as web sites or documents), to become part of a web of information, processes and services where inferences can be drawn, new knowledge gleaned and services provided and consumed at all levels of the XaaS stack.

OpenSEA

We have noted that SOAs need to evolve to include the XaaS landscape. Integration and interoperation are key to this new SOA and ideally it should be capable of including semantic systems within this integration. OpenSEA proposes that any approach needs to be based on open standards in order that enterprises are not divided by proprietary lock-in. The approach needs to be abstract and capable of extension and specialisation to allow different domains to be able to adapt it to their needs. And it needs to combine a common language of business with a common syntax for logic. In the next sections the case will be made for using The Open Group Architecture Framework as the abstract business language, and using ISO24707:2007 Common Logic as the abstract syntax.

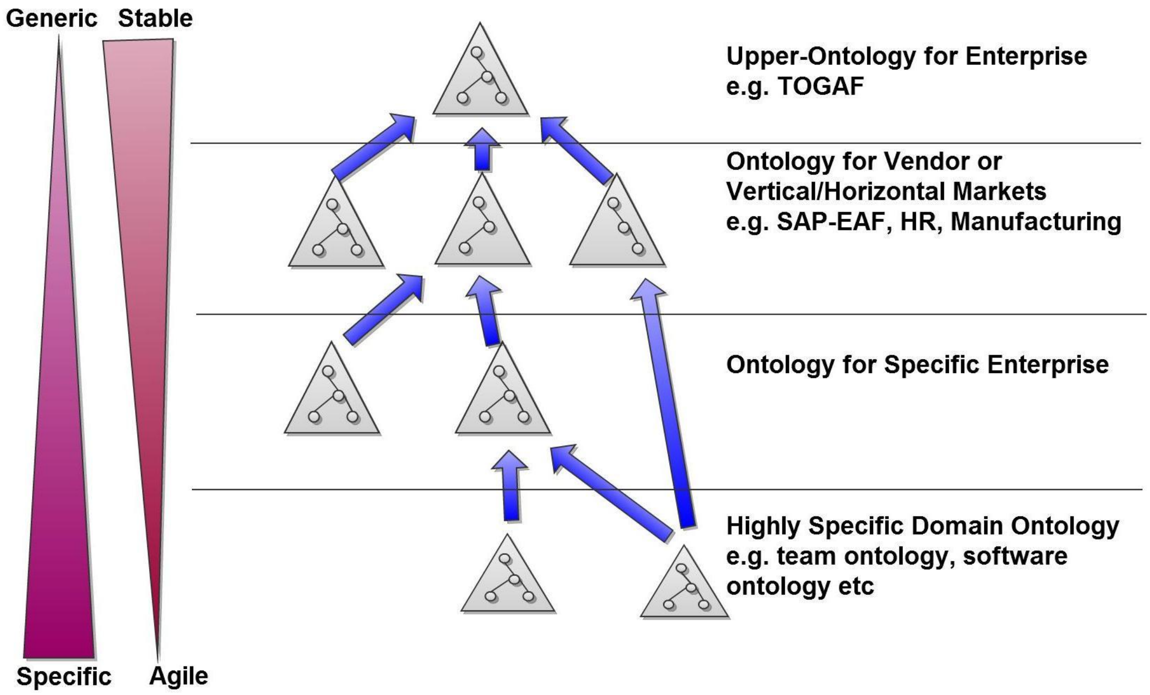

Fig. 2. Specialisation of The Upper Ontology in OpenSEA (from Bridges, 2010).

Figure 2 shows how the upper ontology created by TOGAF is generic yet stable and can be extended and specialised by vendors, markets or enterprises which can in turn be specialised. The different ontologies show how different domains only take the terms that are of use to them and may take them from one or more domains. By retaining a chain of generalisation/specialisation a common generalisation can be found between any two terms/definitions in different domains.

TOGAF – The Upper Ontology for OpenSEA

TOGAF9 is an open and freely licensed framework that is vendor neutral and sector neutral. It was developed by over 300 Architecture Forum members and companies from highly respected IT customers and vendors. It provides a framework for enterprises to extend and specialize, and provides a set of commonly used terms and definitions in the process. Some are formally represented as specific concepts and relations, others as textual descriptions. For example, TOGAF defined ‘Enterprise’ as “any collection of organizations that has a common set of goals. For example, an enterprise could be a government agency, a whole corporation, a division of a corporation, a single department, or a chain of geographically distant organizations linked together by common ownership” (The Open Group, 2009, p. 5) This abstract definition fits within the broad remit of providing an upper ontology that is loose (i.e., general) enough to be meaningful to entire markets, corporations, public bodies, and societies or groups working within these bodies. The Open Group also defined the purpose of Enterprise Architecture in such a way as to meet the requirements of ‘enterprises’ exchanging information openly and meaningfully “to optimize across the enterprise the often fragmented legacy of processes (both manual and automated) into an integrated environment that is responsive to change and supportive of the delivery of the business strategy” (p. 6). Enterprise architecture should therefore provide the platform for innovation and interoperation within and between units of operation of all sizes. It is not limited to information systems. As Tolido (2009) (Vice President of Cap Gemini Netherlands) pointed out Oracle OpenWorld 2009 focused heavily on innovation and being able to reuse existing resources effectively. Indeed, innovation, efficiency, collaboration and cooperation may all be born out of necessity in the cooler economic climate of the early 21st Century.

Because OpenSEA proposes to be a ‘framework’ for interoperation across and within enterprises, it is important to examine how The Open Group (2009) defined an ‘Architecture Framework’:

> a foundational structure, or set of structures, which can be used for developing a broad range of different architectures. It should describe a method for designing a target state of the enterprise in terms of a set of building blocks, and for showing how the building blocks fit together. It should contain a set of tools and provide a common vocabulary. It should also include a list of recommended standards and compliant products that can be used to implement the building blocks (p. 7).

The key points to note here are the common vocabulary and building blocks, which are fundamental components for a semantic market place or semantic interoperation between different enterprises. Note again the convenience of redefining the ‘enterprise’ to go beyond the individual corporations to the concept of a group of corporations trading together within the defined boundary of a market-place. Finally, one of the underlying aims of the TOGAF9 framework is to allow for enterprises to communicate without boundaries. Again, at its heart is the aim to breakdown silos and barriers between discrete units to promote communication and interplay. Information systems operating within architectures that have been guided by TOGAF should therefore experience ‘boundaryless’ information flow. ‘Boundaryless Information Flow’ is a trademark of The Open Group, and represents "access to integrated information to support business process improvements," representing a desired state of an enterprise's infrastructure specific to the business needs of the organization” (The Open Group, (2009, p. 27).

To summarise, The Open Group Architecture Framework has, at its core, many of the implicit semantics required for the integration of disparate and distinct domains. It provides the broad terms and definitions aimed to provide ‘Boundaryless Information Flow’ without specifying any prerequisites or restrictions based on size or market.

ISO 24707:2007 – The Meta-Ontology for OpenSEA

Uschold (2003) identified the major evolutionary paths of the Web as finding resources in an ever growing pool, redefining the Web for human and machine consumption, changing the Web from a pool of resources to a pool of services and semantically enriching those resources. Semantic systems should provide the capability to recognise, represent and react to the meaning of data in the context of the goals of the user (Sowa, 2009, p. 33). Types of semantic systems include deductive databases, expert systems, knowledge based systems and the Semantic Web and its associated applications. However, they can be built on any one of a number of logical languages or formats and interoperation between different systems using different semantic structures or different ontologies can be difficult or impossible. Two legacy systems can be brought to interoperate better than two new, semantically-enabled systems that use different ontologies (Sowa 2009).

As systems start to interact with other systems and corporations look to operate seamlessly with other organisations in an ‘extended enterprise’ (Kuhlin & Thielmann, 2005) this problem becomes global in scale. Common Logic (referred to as CL throughout the Chapter) proposes a standardized approach to develop interoperation between systems using different formalisms and representations. The CL standard outlines its aims as, “The intent is that the content of any system using first-order logic can be represented in this International Standard. The purpose is to facilitate interchange of first-order logic-based information between systems” (ISO/IEC 24707, 2007). It provides a standard for a logical framework for the exchange of data and information across networks, including open networks such as the Internet.

CL dialects must be compliant with the semantics of First Order Logic but CL does not impose any formal syntax, rather it provides an abstract syntax and thereby allows for the reliable translation between languages. The three dialects that currently support CL are;

- CGIF – Provides a serialised representation for conceptual graphs.
- CLIF – A syntax similar to the Knowledge Interchange Format which has become the de facto standard notation for many applications of logic
- XCL – An XML notation for CL that is the intended interchange language for communicating CL across networks.

Common Logic and RDF

ISO/IEC24707:2007, section 5.1.2c, states that “The syntax should relate to existing conventions; in particular, it should be capable of rendering any content expressible in RDF, RDFS, or OWL.” This has been demonstrated by Pat Hayes in the following example (Hayes, 2006).

```
<owl:Class rdf:id="#ChildOfUSCitizenPost1955">
  <owl:intersectionOf rdf:parseType="Collection">
   <owl:Restriction>
    <owl:onProperty rdf:resource="#parentOf" />
    <owl:allValuesFrom>
     <owl:Restriction>
      <owl:onProperty rdf:resource="#isCitizenOf" />
      <owl:hasValue rdf:resource="#USA" />
     </owl:Restriction>
    <owl:Restriction>
     <owl:onProperty rdf:resource="#dateOfBirth" />
     <owl:allvaluesFrom rdf:resource="#YearsSince1955" />
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
```

Maps to

```
(= ChildOfUSCitizenPost1955
    (And (AllAre parentOf (MustBe isCitizenOf USA))
       (AllAre dateOfBirth YearsSince1955) )
```

A further possibility, however, was provided by Hayes (2009) in his keynote speech to a recent International Semantic Web Conference 2009, in which he showed that RDF is almost Peircian in notation, and how RDF Redux theme could become a fully expressive CL dialect. Doing so would allow integration with other CL compliant dialects and greatly simplify the semantic stack (or ‘layer cake’). Figure 3 shows how RDF simplifies the semantic stack of tools and protocols.

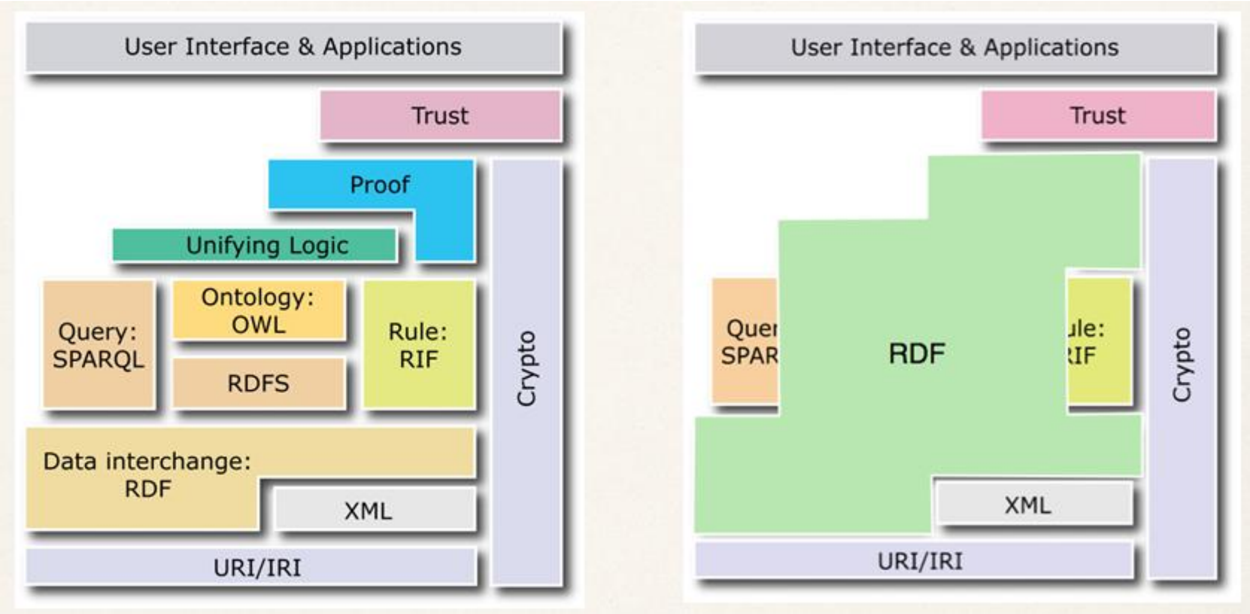

Fig. 3. Hayes' Vision of RDF within Common Logic (Hayes, 2009).

OpenSEA -- Developing an upper ontology with CL

By expressing the terms and definitions used by TOGAF in a CL compliant format, and allowing these definitions to be extended and specialized freely, OpenSEA aims to create the abstract framework required for different domains to exchange information without requiring a rigid ontology nor a specific system or language.

In this chapter we use Conceptual Graphs which can be expressed in a CL dialect CGIF. Graphical notation is used as the primary representation to express how full logic can be portrayed in a format that is readily consumed by human agents yet can also be expressed in a compact linear notation that can be converted to any other CL compliant dialect without loss of meaning. The graphs were developed using CharGer (sourceforge.net/projects/charger), Amine (sourceforge.net/projects/amine-platform) and CoGui (www.lirmm.fr/cogui/).

Interlinked Domains

Key to the framework is the need for a chain of generalization or specialization in order for two disparate domains using different ontologies to come to an common understanding. one can imagine the chain as one of simple ‘IS-A’ relationships where all concepts and relations ultimately link back to a definition in the upper ontology as provided by TOGAF. The approach is similar to the idea of the Domain Naming System which relies on lookups and, at its heart, has the ‘.’ domain above all others to provide the link between top level domains.

John Sowa (via email private correspondance) has provided the following examples of linear expression for this generalisation/specialisation concept, as shown below.

```
CLIF:
(forall ((R1 MonadicRelation) (R2 MonadicRelation) (x) (y))
 (if (and (GeneralizationOf R1 R2) (R2 x y)) (R1 x y)))

CGIF:
[MonadicRelation @every *R1] [MonadicRelation @every *R2]
[Entity: @every *x] [Entity: @every *y]
[If (GeneralizationOf ?R1 ?R2) (#?R2 ?x ?y) [Then (#?R1 ?x ?y)]]
```

This says that for all monadic relations R1 and R2 and any x and y, if R1 is a generalization of R2 and R2(x,y), then R1(x,y). Once the GeneralisationOf statement is made then the type hierarchy can be listed as a simple collection of assertions:

```
CLIF;
 (and (GeneralizationOf Architect Business_Analyst)
  (GeneralizationOf Architect Information_Analyst)
  (GeneralizationOf Information_Analyst Data_Analyst)
  (GeneralizationOf Information_Analyst Technical_Analyst))
```

For example, in TOGAF the relation ‘Performs Task In’ is formalized as a canon (common usage) as relating an ‘Actor’ to a ‘Role’. Suppose a health-specific domain specializes the term ‘Actor’ to cover the concepts of ‘Doctor’ and ‘Patient’ whilst a Sales and Distribution domain specializes the same term to concepts such as ‘Salesperson’, ‘Lead’ and ‘Customer’. If the domains were required to interact, for example a pharmacy needed to purchase drugs, the two domains could agree on the fact that a ‘Doctor’ and a ‘Salesman’ share some commonality in that they both perform tasks in their respective roles.

This example can be expressed as follows:

```
TOGAF:
[Actor: @every *t]
 (PerformsTaskIn ?t [Role])
HealthCare
(GeneralizationOf Agent Doctor)
(GeneralizationOf Role Healthcare)
Sales
(GeneralizationOf Agent Salesman)
(GeneralizationOf Role Sales)
```

that we can translate to the CLIF form:

```
CLIF:
[Doctor: @every *t]
 (PerformsTaskIn ?t [Healthcare])
And
[Salesman: @every *t]
 (PerformsTaskIn ?t [Sales])
```

Any logic or inferences that can be made in the TOGAF domain would be equally expressed in both the sub domains and any knowledge farms that can make those generalised rules has the capacity to gather new information from data collected from disparate domains.

It is worth noting that unlike DNS, OpenSEA relies on the fact that the specialization and generalization is not a simple hierarchy but allows for concepts to be specialisations of one or more ‘master’ concepts. The classic example is ANGEL (Sowa, 1984, p. 408) where ANGEL is a specialisation of both ANIMATE and MOBILE-ENTITY. This example also includes the specification that ANGEL < ¬PHYSOBJ ie an angel is not a physical object, the IS-NOT specification being a powerful tool for future developments of OpenSEA.This notion of specialisation extends from concepts to instances. For example, Doctor Bob Smith is a keen motorsport fan. Within the health domain a patient is unlikely to be interested in which team Dr Smith follows so the information is superfluous. Yet a domain specializing in providing executive travel to Formula 1 events may be very interested in the fact that the instance ‘Bob Smith’ IS-A ‘Doctor’ and Bob Smith ‘IS-A’ ‘McLaren fan’ if they have an inference engine that could deduce that ‘if a fan performs tasks within certain roles then the fan has disposable income’.

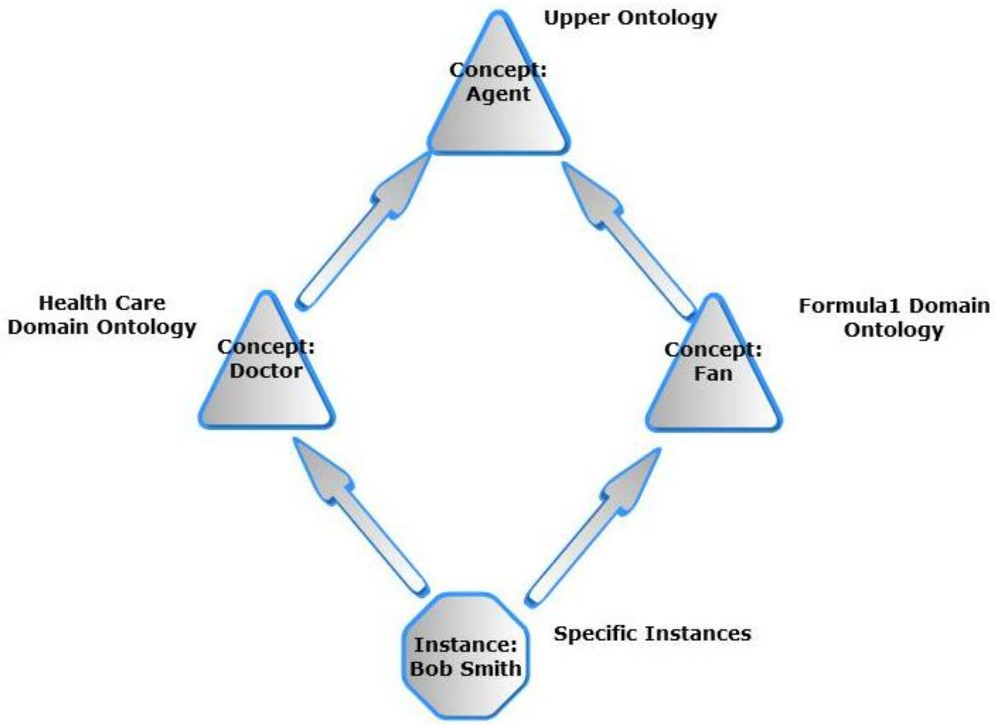

Fig. 4. Simple example of An Instance Being A Specialisation of Different Concepts

Developing an Upper Ontology from TOGAF

This section draws on the research of (Bridges, 2010; Bridges & Polovina, 2010). One of the problems faced was the task of viewing terms abstractly, even if they are as well defined as those in the TOGAF9 material. Bridges observed the problem lay with disengaging from the precepts that any individual holds and the assumptions that shape how an individual perceives the world.

Sowa's Conceptual Catalogue

Bridges referred extensively to the conceptual catalogue created by Sowa (1984, pp. 405-424) (herein referred to as SCC) which provides a number of 'canons' (i.e., common meanings) for a broad range of terms. However, the real value of the catalogue is in what has been canonised as well as how it has been described. It is possible to see how a few high level concepts and relations can provide a broad range of conceptual structures and move towards an unbiased and abstracted definition (or canon) for terms that are so common they are hard coded into the mind of the individual without any clear and logical structure.

By reusing the SCC OpenSEA adheres to the principle advocated by Berners Lee and Kagal (2008) to not reinvent the wheel and to use existing ontologies wherever possible. It is also hoped that using something as established as the SCC would provide some base commonality with other ontologies that have also been influenced by it.

The broad concepts of many enterprise architectures (What, Where, When, How, Why and Who) provide a useful approach to determining upper level concepts as high level contexts. This is mirrored to some extent in the SCC. For example, the context of 'What' could map to Sowa's ENTITY concept and specialisations of this concept would all have a common general meaning, whether they are Data Entities or Abstract Objects.

Similarly, a high level concept of 'how' could provide the specialisations of Process, Business Function, Business Service, Information System Service etc (again, examples taken from the TOGAF9 definitions).

This approach is shown in Figure 5 as part of an initial concept type hierarchy.

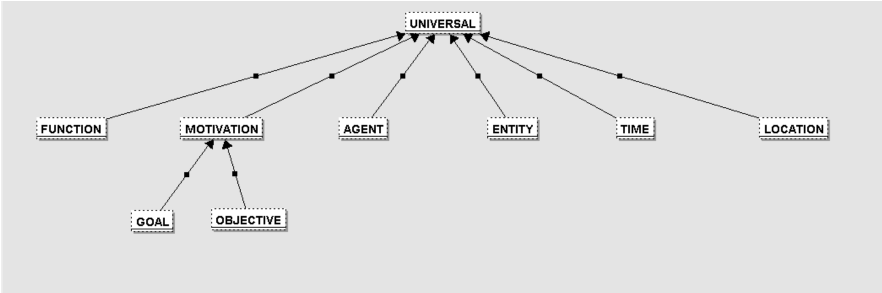

Fig. 5. An Initial Type Hierarchy for TOGAF Terms (Bridges, 2010)

TOGAF describes Data Entity as, "an encapsulation of data that is recognized by a business domain expert as a thing. Logical data entities can be tied to applications, repositories, and services and may be structured according to implementation considerations" (The Open Group, 2009).

In other words, it too is covered by Sowa's canon of 'physical objects as well as extractions'. If we continue to focus on the elements of an enterprise, resources and agents can be seen as specialisations of ENTITY within the SCC. Events are also present within the SCC. Events could be seen as being part of a process involving different states (another SCC concept) and it is easy to see how 'who', 'how' and 'when' could be linked to a process.By including the concept of 'why' the business goals and objectives from TOGAF9 are brought within the Architecture framework.

Figure 6 builds on the basic concepts of Figure 5 by adding the EVENT concept. It also adds STATE as this is an integral part to a process that should be involved with changing the state of an entity.

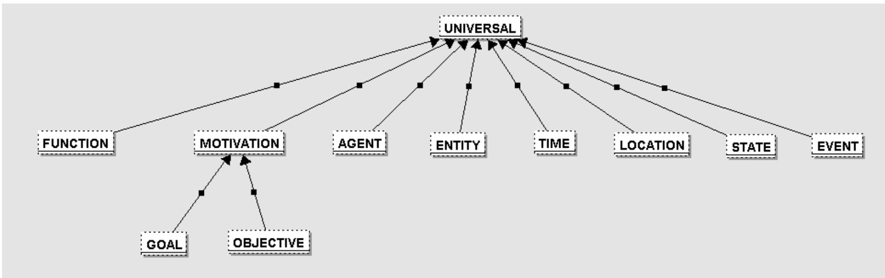

Fig. 6. Development of the Type Hierarchy.

Within the SCC Sowa described 'CHARACTERISTIC' (a specialisation of the concept ATTRIBUTE) as being 'essential in nature'. By this canon the object attributes shown in Table 1 can also be seen as characteristics, a confusion that needs to be addressed in the ontology if the SCC is to be used to any extent.

Table 1. TOGAF Attributes for all Metamodel Objects

Metamodel Attribute	Description
ID	Unique identifier for the architecture object.
Name	Brief name of the architecture object.
Description	Textual description of the architecture object.
Category	User-definable categorization taxonomy for each metamodel object.
Source	Location from where the information was collected.
Owner	Owner of the architecture object

Attributes that are essential could be linked to their associated objects by a 'chrc' relation to denote the fact that they are essential in describing individuals of those concepts and could provide a very useful means of defining individuals within a global market place. For example, all metamodel objects within an enterprise must have the attributes of ID, Name, Description, Category, Source and Owner (Table 1). These same attributes could be used to provide the information needed to help identify resources, services, processes etc within an open market and provide the commonality to assist with interoperation. This approach the expansion of Sowa's type hierarchy (Figures 7a and 7b).

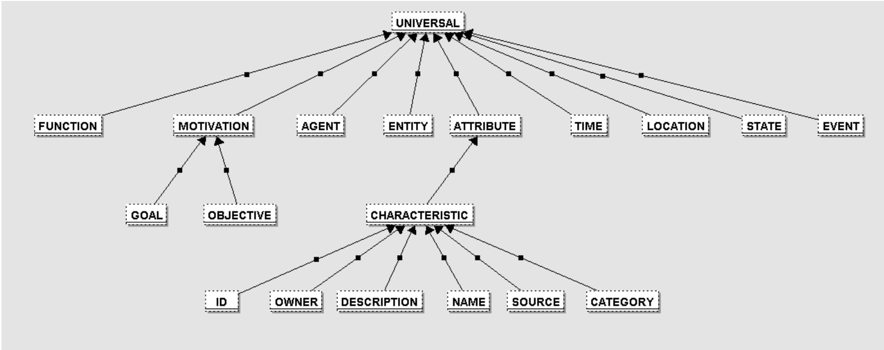

Fig. 7a. Common Attributes Within The Type Hierarchy.

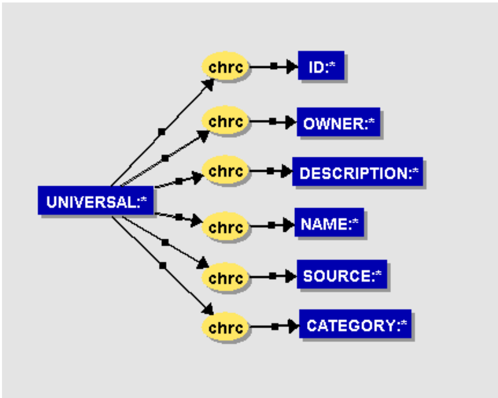

Fig. 7b. A Conceptual Graph Representation for all Objects and Their Minimal Metadata.

In CGIF, Figure 7b would be represented as

```
[Universal: @every *t]
 (chrc ?t [Category])
 (chrc ?t [Description])
 (chrc ?t [ID])
 (chrc ?t [Name])
 (chrc ?t [Owner])
 (chrc ?t [Source])
```

In the framework the ID, Category, Source and Owner are all represented using URLs, and the name and description by a human-readable text string. The attributes are extended with ‘Definition’, which provides for a CL based definition of how the object is defined by other objects. ‘Category’ is used within OpenSEA as the means of embedding the well recognized ‘IS-A’ relationship..

Knowledge bases can harvest this information and build links between different domains using the shared generalisations and generate new information. These ‘knowledge farms’ could be the pioneers of the new, distributed architecture by acting as both brokers (identifying what services and processes are available and those that are in demand) and offering to integrate the enterprise business rules engine within the larger market place.

Modelling Relationships Within TOGAF9

Table 2 shows some of the relationships that can occur within the TOGAF9 metamodel. Relationships can also be captured in CG as shown in Figures 8a and 8b, in this example the relationship ‘Resolves’ is shown with a signature consisting of an Actor and an Event, as is the relationship that an Actor ‘generates’ an Event.

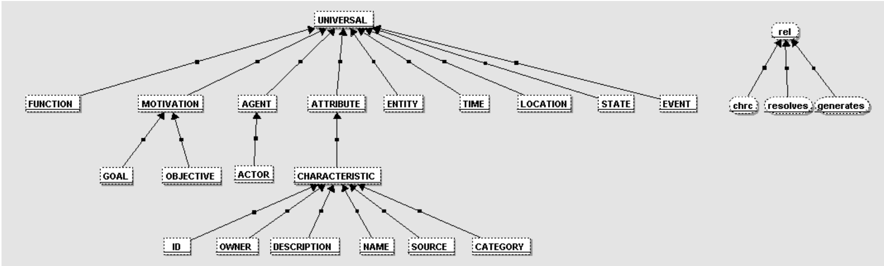

Fig. 8a. Capturing Relationships within the Type Hierarchy.

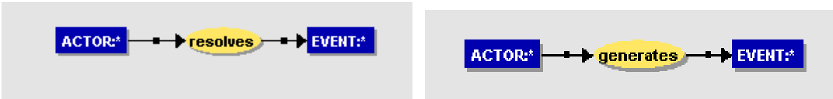

Fig. 8b. Graphical Representations of Simple Canons.

Nested graphs

Zachman and Sowa (1992) showed that concepts could be seen as nested graphs so that, for example, ‘how’ something is achieved is represented using symbols relevant to the agent at that level. In Figure 9 Zachman and Sowa illustrated the difference between how agents operating at the Enterprise Level would view ‘what’ in terms of an entity where as those operating at the level of Information System analysis would regard ‘what’ in terms of data yet the two contexts are related by a relation (‘NAME’). Similarly, business process experts would consider the business process when considering ‘how’ something is done but this has to be mapped to the system analyst’s contextual view in terms of the functions called within the system. In this case the two are connected by the ‘MODL’ relation.

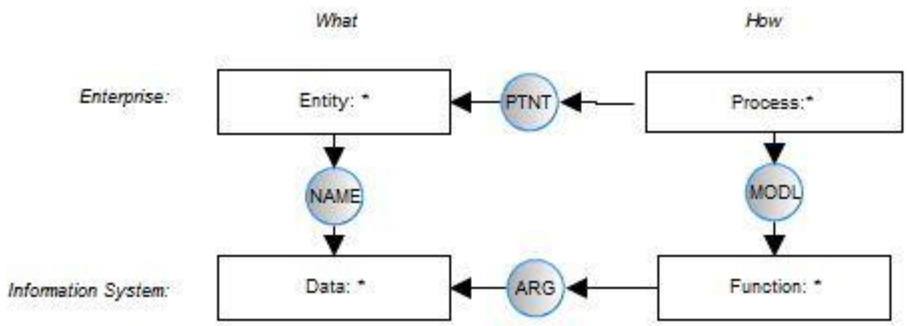

Fig. 9. Inter-related Contexts (from Zachman and Sowa, 1992, p. 610).

As already identified, TOGAF9 models the architectures of Business, Information (Architecture and Data) and Technology. Each tier of the TOGAF model could be considered to be a context in a similar fashion to Zachman’s and Sowa’s formalisation of the ISA . Within the Business Architecture a Business Service (‘what’) could be a unit that is ‘owned and governed by’ an Organisational Unit (‘who’) which in turn ‘operates in’ a location (‘where’). From a System Analyst perspective the same Business Service ‘provides’ or ‘consumes’ a Data Entity (‘what’) which ‘resides within’ a Logical Data Component (which could be argued to be an abstract object, i.e. another entity) which is, in turn ‘realised by’ a Physical Data Component (another entity). Figure 10 shows the type hierarchy and relations being developed for this purpose.

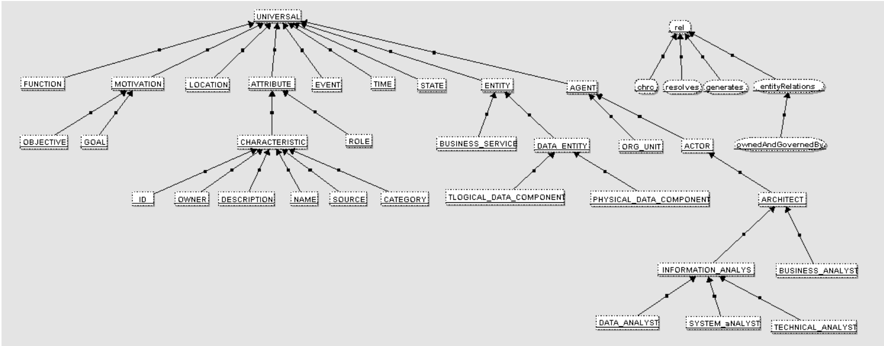

Fig. 10. TOGAF Objects and Relations Captured in the Type Hierarchy.

Contextualisation of Information

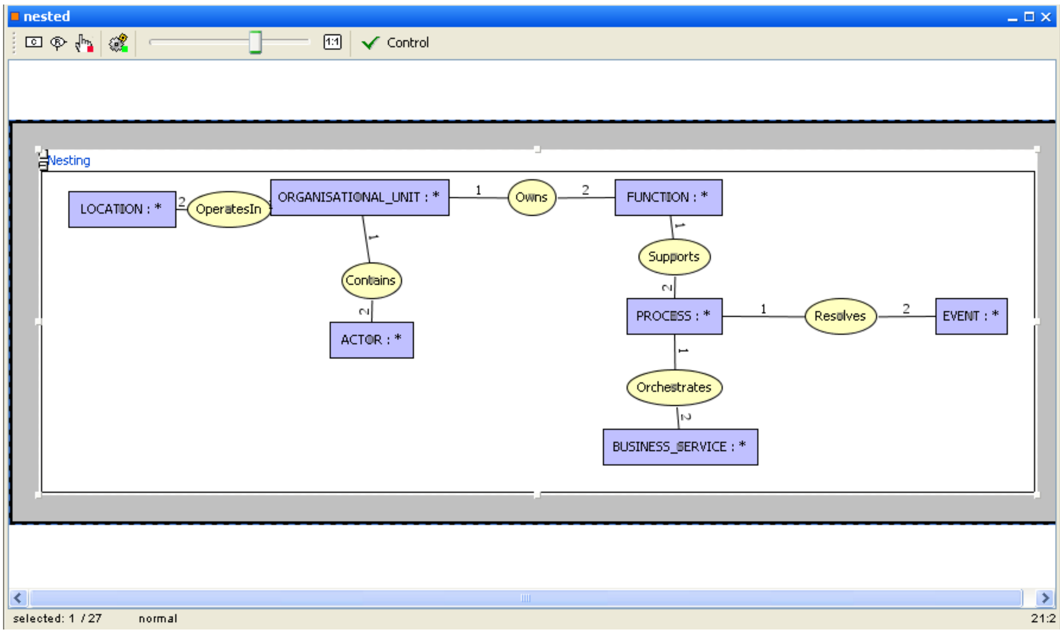

Fig. 11. 'Business Service' Modelled Within the Perception of a Business Analyst (Bridges, 2010).

Figure 11 shows a nested graph within the concept 'Business Analyst' and illustrates the contextual meaning of a Business Service to a Business Analyst (i.e. an agent working at the layer of business architecture). The types and relations are highlighted in Figure 12 which focuses on the agents and relationships within the type hierarchy.

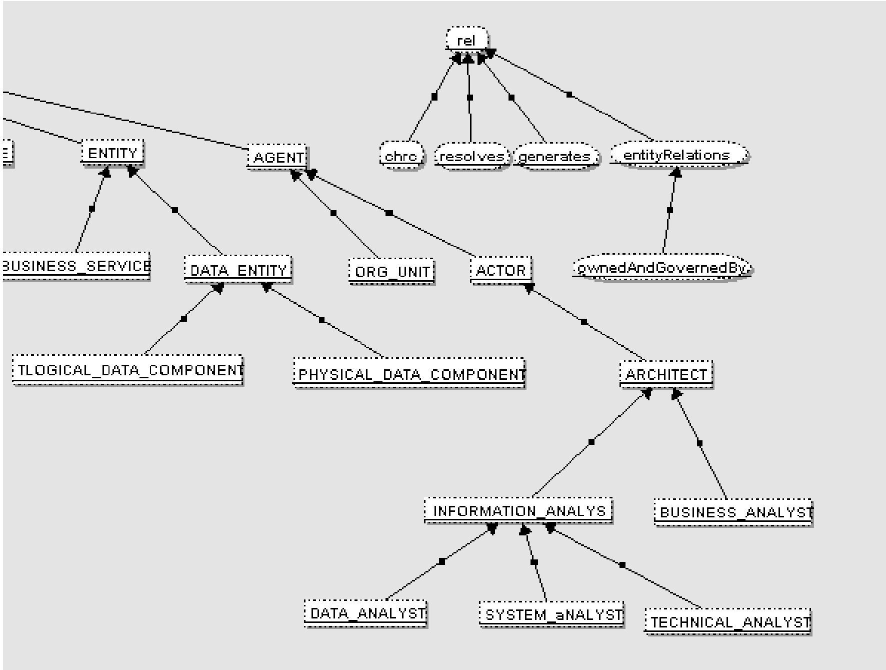

Fig. 12. Focus on the Relations and Actors In the Type Hierarchy.

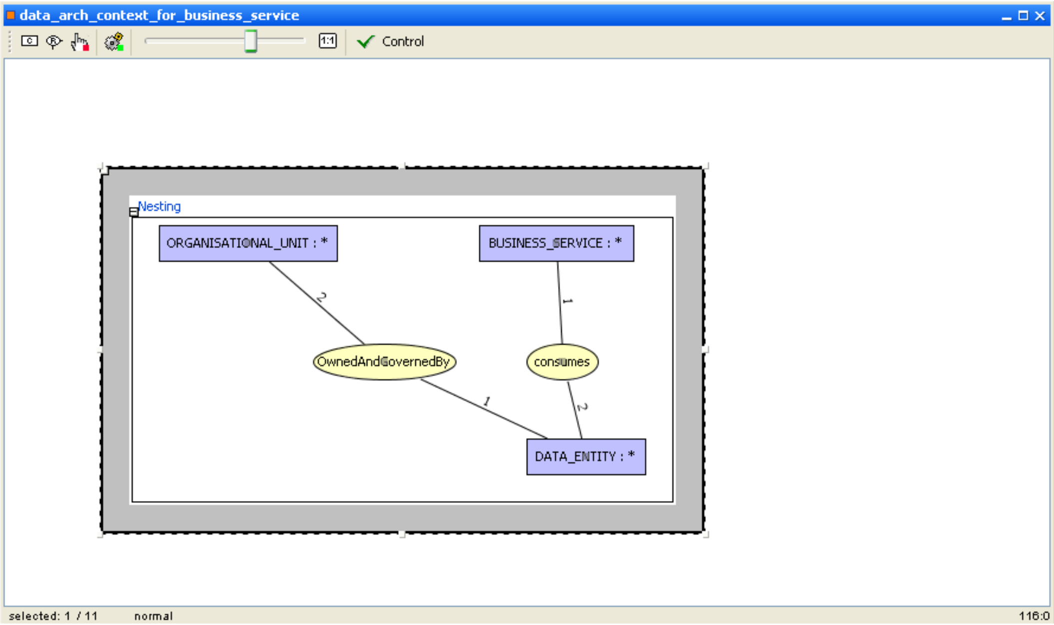

Fig. 13. 'Business Service' as Perceived by a Data Analyst (Bridges, 2010).

In Figure 13 the concept of a business service is nested within the context of a Data Analyst to show what a Business Service means to an agent operating within the Data Architecture layer of an enterprise.

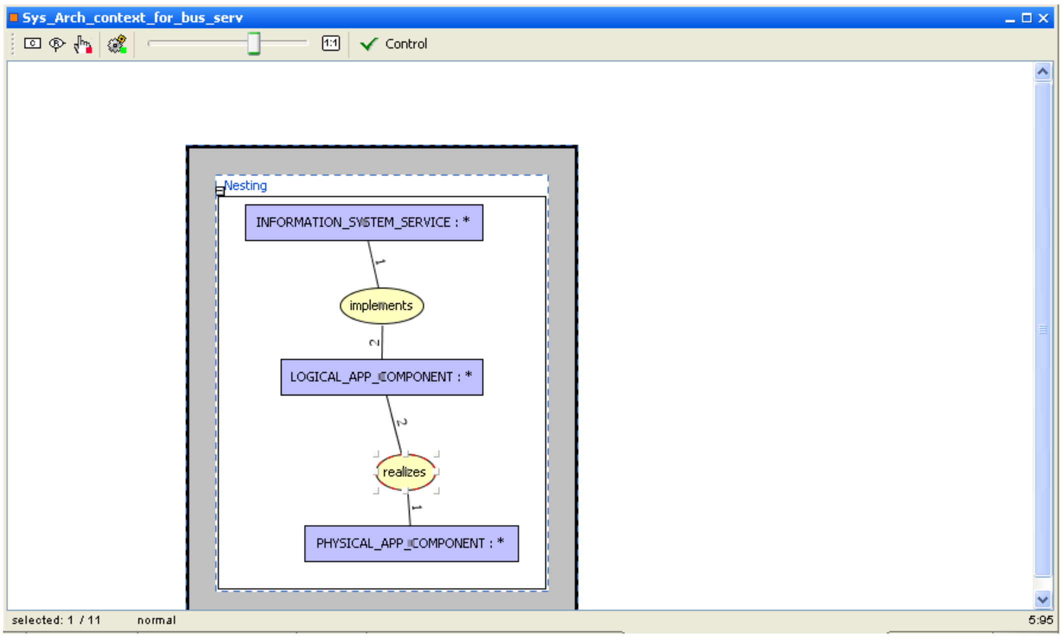

Fig. 14. 'Business Service' As Perceived by a Systems Analyst (Bridges, 2010).

In Figure 14 the context of a business service is shown as per a Systems Analyst. The Business Service is shown as the more specialised 'Information System' inferring that it is a fully automated business service (a fact that would need representing in the type hierarchy).

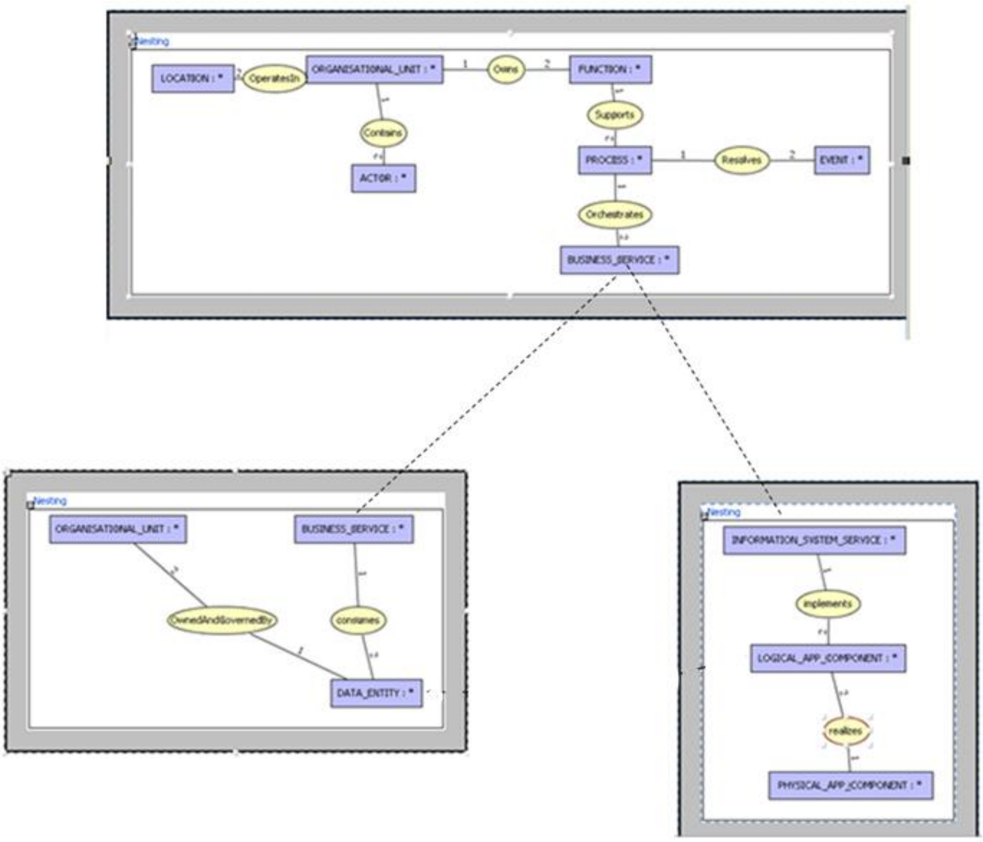

Fig. 15. Different Perceptions Modelled With Co-Referents (Bridges, 2010).

Figure 15 shows the three different nested graphs representing the different contextual perceptions of what a business service means interlinked by co-referents (the dotted lines). The co-referents allow the three agents to interconnect and for changes in one tier to be integrated within the other tiers as a single version of the truth.

Relationships between different nested graphs could be used to connect different contexts and so this simple example shows how the different interpretations for what a concept means to different agents can be interlinked.

OpenSEA and the Cloud

The TOGAF semantics may be used to describe the XaaS offerings of the cloud by compartmentalizing the services between the external face and inner workings. Consumers of the services are interested in different aspects of the service to those that run it, as shown in Figure 16.

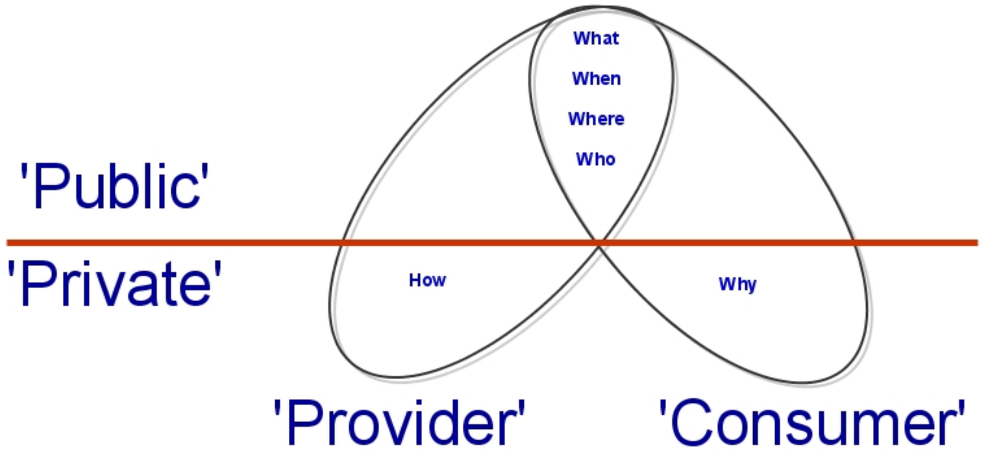

Fig. 16. The Public/Private Differentiation of XaaS (Bridges, 2010).

Figure 16 shows that the provider is 'privately' interested in 'how' the service is provided whilst the consumer has little concern of 'how' the service is realised. Within TOGAF this would relate to the lower tiers and the contexts relevant to software engineers, hardware engineers, etc. TOGAF semantics used at this level could include 'Physical Technology Component' 'is hosted in' a 'Location' and 'Realizes' a 'Physical Application Component'. Similarly, a 'Service' 'is Realized through' a 'Logical Application Component' and is 'Implemented on' a 'Logical Technology Component'

The consumer has their own individual objectives and goals related to the 'why' a service should be used and these may vary from consumer to consumer and is of little or no interest to the provider. TOGAF terminology relevant to the consumer but not the provider could be a 'Goal' 'is realized through' an 'Objective' or 'Addresses' a 'Driver'.

The terms used will be determined by the service provided, for example Hardware as a Service is more interested in the physical aspects of the infrastructure such as RAM, storage and processing power, infrastructure may include all this plus virtual machines, operating systems, software and scalability. Platform as a Service offers the chance to build and deliver web applications but may not have any reference to the underlying hardware or infrastructure.

The *consumption* of a service, however is dependent on 'what', 'when', 'where' and 'who' and consumers and providers could advertise their need/provision of the service in these terms for brokers to 'matchmake' or agree contracts. TOGAF terms that would be used in agreeing the contract could include a 'Service' 'Provides' or 'Consumes' a 'Data Entity' and 'Provides Governed Interface Access' to a 'Function' whilst 'Service Quality' 'Applies to' a 'Contract' and 'is Tracked Against' a 'Measure'.

This shift in emphasis from service provision to transaction fulfilment could be referred to as a step away from *Service* Oriented Architecture to *Transaction* Oriented Architecture where the provision and consumption of services are perceived as part of a whole. After all, a Service means nothing without a consumer; it is defined by consumption.

OpenSEA and Web3.0

In the introduction reference was made to the advantages inherent in users and consumers being able to capture what things are and the relationships between them (Mulholland 2009). If web sites were annotated with either XCL or the less verbose CLIF or CGIF strings (using XML tags to identify the strings) the user could use simple client tools to express the inter-relations graphically and visualize how the web-site sits within the greater web. A web cache could become a powerful information set, moving beyond a series or URL's to a web of knowledge and guided reference. Similarly, search engines could gather XCL or other CL dialect from any Web sites or data sets that support OpenSEA to create powerful semantic searches, broker services or generate new knowledge through rules engines.

OpenSEA and the Software Engineer

Trapp (2009) outlined how Semantic Web technologies can can add some elements of Knowledge Management to enterprise software. By introducing the expert's knowledge to the data and functions within a software the entire system (by which we mean the human agents and the software) can become 'intelligent'. This can be incorporated within the software through metadata (e.g. XCL definitions of business objects or processes), reasoning and visualisation amongst others.

Trapp refered to the 'Design Time Type Information' Open Source project available under Apache. DTTI combines a base ontology with REST web services that expose RDF data about objects within SAP systems. He suggests some of the benefits include formalising the architecture, using reasoning to detect direct and indirect dependencies and forbidden dependencies. All this is possible in the OpenSEA framework.

Example

Within TOGAF9 (2009) a Data Entity has the following interactions with its environment:

Table 2. Concepts and Relations Referring to 'Data Entity'

Source Object	Target Object	Relationship
Data Entity	Logical Application Component	Is processed by
Data Entity	Logical Data Component	Resides within
Data Entity	Service	Is accessed and updated through
Data Entity	Data Entity	Decomposes
Data Entity	Data Entity	Relates to

Added to the Type Hierarchy, we now have the following.

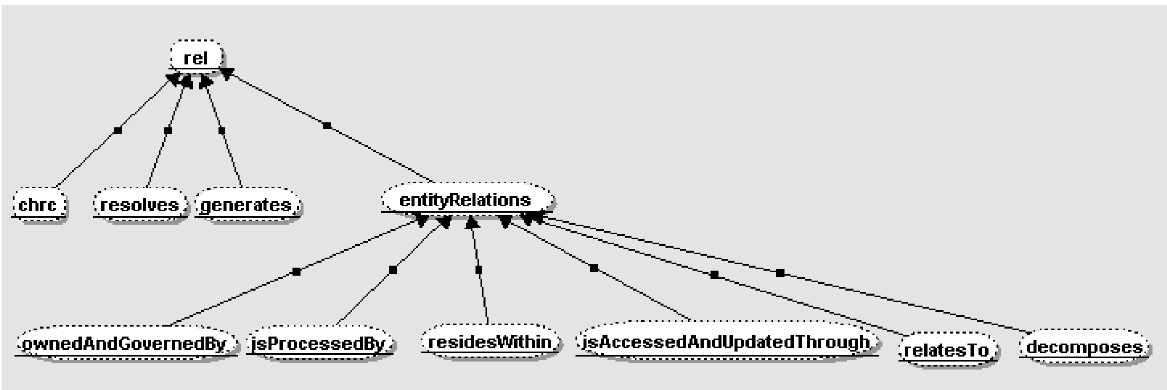

Fig. 14. Relations for Data Entity captured within a Type Hierarchy

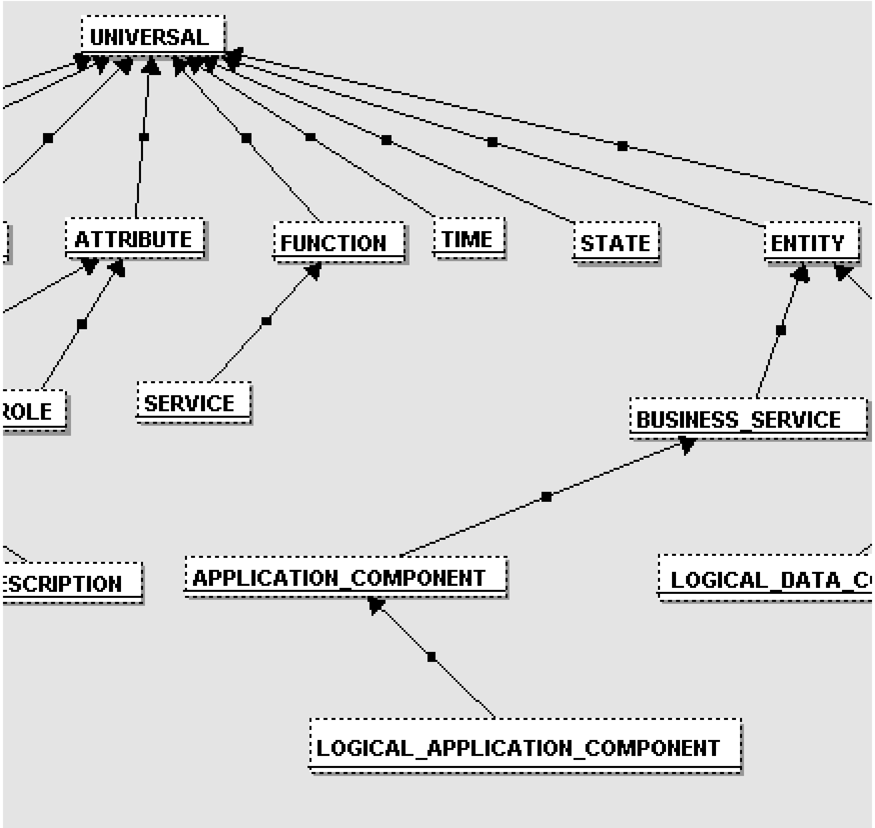

Fig. 15. Specialisations of the Concept 'ENTITY'

Data Entities are stored within the knowledge base of all participating domains as:

CGIF:
[DEFINITION: "[DATA_ENTITY:*x1] [SERVICE:*x2] (isAccessedAndServicedThrough ?x1 ?x2) "] [NAME: Data Entity] [CATEGORY: OpenSEA.org/Universal] [SOURCE: "http://www.opengroup.org/architecture/togaf9-doc/arch/index2.html"] [ID: "OpenSEA.org/DATA_ENTITY"] [OWNER: OpenSEA] [DESCRIPTION: "AN ENCAPSULATION OF DATA..."]
[DATA_ENTITY: *x1]
(chrc ?x1 OpenSEA) (chrc ?x1 Universal) (chrc ?x1 "http://www.opengroup.org/architecture/togaf9-doc/arch/index2.html") (chrc ?x1 "AN ENCAPSULATION OF DATA...") (chrc ?x1 "OpenSEA.org/DATA_ENTITY") (chrc ?x1 Data Entity) (chrc ?x1 ?x1] [SERVICE:*x2] (isAccessedAndServicedThrough ?x1 ?x2) ")

In the example, the generic DATA_ENTITY contains its own DNA containing examples of how it can be used, (in the DEFINITION), what it's a specialisation of (CATEGORY), a plain text description, the URL at which it is defined and so on. Any specialisations of the generic DATA_ENTITY refers to this ID as its category, thereby retaining the links that make up the web of references.

Further Investigations

Integration of OpenSEA and GoodRelations

GoodRelations (Hepp, 2008) provides a standard vocabulary for expressing services and products that are offered on web sites. OpenSEA should not be seen as an alternative to an established ontology as the aims are similar but the ontology of GoodRelations could be investigated as a specialization of TOGAF. Furthermore, Hayes (2009) suggestion that RDF has the capacity to become CL compliant means the RDF expression of GoodRelations could be expressed in XCL or as a CLIF string within XML brackets. This could introduce the potential for users of GoodRelations to integrate with other domains of knowledge through OpenSEA and human consumers of web-sites could use simple clients to visualize how the information on the site relates to the greater Web.

Knowledge, Inference and Information Generation through OpenSEA

OpenSEA can, by definition, be used as part of an information generating rules engine through the application of rules (knowledge) on the information available to infer new, un-tapped information or making decisions on new courses of action.

Sowa has reported that SBVR is capable of full CL syntax and as such it could be included as part of the OpenSEA framework by providing the meanings used in Business Rules. The overlap between SBVR and Controlled Natural Language could act as a bridge between human and machine agents by bringing the two representations closer together. For example, (Baisley et. al., 2005),

> Below is a description of the semantic formulation of the rule above expressed in terms of the SBVR Logical Formulations of Semantics Vocabulary. It is easily seen that SBVR is not meant to provide a concise formal language, but rather, to provide for detailed communication about meaning. The description is verbose, when separated into simple sentences. But it captures the full structure of the rule as a collection of facts about it.
>
> > The rule is meant by an obligation claim.
> > That obligation claim embeds a logical negation.
> > The negand of the logical negation is an existential quantification.
> > The existential quantification introduces a first variable.
> > The first variable ranges over the concept ‘barred driver’.
> > The existential quantification scopes over a second existential quantification.
> > That second existential quantification introduces a second variable.
> > The second variable ranges over the concept ‘rental’.
> > The second existential quantification scopes over an atomic formulation.
> > The atomic formulation is based on the fact type ‘rental has driver’.
> > The atomic formulation has a role binding.
> > The role binding is of the fact type role ‘rental’ of the fact type.
> > The role binding binds to the second variable.
> > The atomic formulation has a second role binding.
> > The second role binding is of the fact type role ‘driver’ of the fact type.
> > The second role binding binds to the first variable.
>
> Note that designations like ‘rental’ and ‘driver’ are used above to refer to concepts. The semantic formulations involve the concepts themselves, so identifying the concept ‘driver’ by another designation (such as from another language) does not change the formulation.

Conclusion

OpenSEA is a framework for unified modelling tools for enterprise architecture. Zachman and Sowa (1992) provided examples of how Conceptual Graphs could be used to model Zachman’s Information System Architecture. The same ideas are contained within the OpenSEA framework as it seeks to formalise architectures that are aligned with TOGAF using CL compliant dialects which, of course, includes CGIF which has a graphical representation.

By formalising the links between the horizontal components (in the ISA these include Enterprise Model, System Model, Technology Model and Component Level, similar to the tiers of TOGAF9) and vertical components (What, How, Where etc) of the ISA Zachman and Sowa identified that each unit could be represented with respect to the other units and each unit could be represented using graphical CGs which had the power to embed full predicate calculus within an easily accessible form.

It is worth noting some quotes taken from Sowa's and Zachman's paper. The quotes continue to be relevant today and hold particular significance when viewed in light of the findings of the International Research Form 2008:

> Dramatic improvements in the price-performance of information technology and the escalation of the rate of change show no signs of abatement. In the words of Alvin Toffler, "Knowledge is change . . . , and accelerating knowledge, fuelling the great engine of technology, means accelerating change." Gone are the days of computers for simple calculations. We are only now beginning to see the enormous complexity of integrating information technology into the very fabric of our enterprises. *Soon, the enterprise of the information age will find itself immobilized if it does not have the ability to tap the information resources within and without its boundaries* [italics ours] (Zachman & Sowa, 1992, p.613)..
>
> an enterprise will form into a free market structure if the nature of the transaction between two organization units is simple, well defined, and universally understood. In this case, the organization (or person) with work to assign would survey all possible workers to find one who is acceptable in terms of availability and cost. This method is much like a stock buyer who scans the pool of stockbrokers to find one who will execute a buy within an agreeable time and for a reasonable fee (Zachman & Sowa 1992, p. 596).
>
> Tools exist for both users and developers. Tools such as online help are there specifically for users, and attempt to use the language of the user. Many different tools exist for different types of developers, but they suffer from the lack of a common language that is required to bring the system together. It is difficult, if not impossible, in the current state of the tools market to have one tool interoperate with another tool (The Open Group, 2009, p.418).

It is worthwhile noting that if the nature of the dependency between cells could be understood and stored in the repository along with the cell models, it would constitute a very powerful capability for understanding the total impact of a change to any one of the models, if not a capability for managing the actual assimilation of the changes (Zachman & Sowa, 1992, p.603).

www.Open-SEA.org

The domain Open-SEA.org has been created to facilitate open discussion and development of the ideas generated in this chapter. Please email shaun.bridges@Open-SEA.org for more information.

References

Baisley, D.E., Hall, J., Chapin, D. (2005). Semantic Formulations in SBVR. Paper presented at W3C Workshop on Rule Languages for Interoperability. Washington, D.C.: April 27-28. Retrieved December 15, 2010, from http://www.w3.org/2004/12/rules-ws/paper/67/

Berners-Lee, T. & Kagal, L. (2008). The fractal nature of the Semantic Web. AI Magazine, 29(3), 29.

Boisot, M, & Canals, A. (2007). Data, information, and knowledge: Have we got it right?. In M. Boisot, I. MacMillan, K. S. Han, K. S. (Eds.), *Explorations in Information Space: Knowledge, Agents, and Organization* (pp. 15-47). Oxford: Oxford University Press.

Bridges, S. (2010). *The Extent and Appropriateness of Semantic Enterprise Interoperability with TOGAF9 and ISO Common Logic.* Unpublished Dissertation, Sheffield Hallam University, Sheffield, UK.

Bridges, S. & Polovina, S. (2010). An OpenSEA framework using ISO24707 common logic. In: F. Xhafa, D. Stavros, C. Santi, & A. Ajith (Eds.), *2nd International Conference on Intelligent Networking and Collaborative Systems, Thessaloniki, Greece, November 24-26: Proceedings*, (pp. 335-336). Los Alamitos, CA: IEEE Computer Society.

Cloud Computing Use Case Discussion Group. (2010). Cloud Computing Use Cases, Version 4.0. Retrieved December 20, 2010, from http://opencloudmanifesto.org/Cloud_Computing_Use_Cases_Whitepaper-4_0.pdf

Hayes, P. (2006). IKL presentation for Ontolog. Retrieved December 21, 2010, from http://www.slideshare.net/PatHayes/ikl-presentation-for-ontolog [Accessed October 18, 2009].

Hayes, P. (2009, November 24). BLOGIC or Now What's in a Link?. Retrieved December 22, 2010, from http://videolectures.net/iswc09_hayes_blogic/

Hepp, M, (2008). GoodRelations: An ontology for describing products and services offers on the Web. *16th International Conference on Knowledge Engineering and Knowledge Management, Acitrezza, Italy, September 29 - October 3: Proceedings,* LNCS 5268, (pp. 332-347). Berlin/Heidelberg: Springer-Verlag.

Heuser, L., Alsdorf, C. & Woods, D. (2009). *International Research Forum 2008,* 1st ed., New York: Evolved Technologies Press.

ISO/IEC 42010 (2007). Recommended practice for architectural description of software-intensive systems. Retrieved October 16, 2009, from http://www.iso.org/iso/catalogue_detail.htm?csnumber=45991

ISO/IEC 24707. (2007). Common Logic (CL): A framework for a family of logic-based languages. Retrieved November 1, 2009, from http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html

Kuhlin, B. & Thielmann, H. (2005). *The Practical Real-Time Enterprise,* Berlin/Heidelberg: Springer-Verlag. Retrieved October 16, 2009, from http://www.springerlink.com/index/10.1007/b138980

MacManus, R. (2009). Eric Schmidt Defines Web 3.0. Retrieved October 20, 2009, from http://www.readwriteweb.com/archives/eric_schmidt_defines_web_30.php

Mulholland, A. (2009). Time to return to the Semantic Web again! | CTO Blog | Cap Gemini | Consulting, Technology, Outsourcing. Retrieved October 21, 2009, from http://www.capgemini.com/ctoblog/2009/07/time_to_return_to_the_semantic.php

Muholland, A. (2010, Sept. 6.). Genuine progress on clouds September 6, 2010. Retrieved December 12, 2010 from http://www.capgemini.com/ctoblog/uncategorized/genuine_progress_on_clouds-php/

Pascale, R.T. (2000). *Surfing the Edge of Chaos: The Laws of Nature and the New Laws of Business*, New York: Crown Business.

Sowa, J. (2009). Controlled natural languages for semantic systems - A roadmap of directions to explore. Retrieved October 21, 2009, from http://www.jfsowa.com/talks/cnl4ss.pdf

Sowa, J.F. (1984). *Conceptual Structures: Information Processing in Mind and Machine.* Reading, Mass: Addison-Wesley.

The Open Group (2009). *TOGAF Version 9*, Zaltbommel, Netherlands: Van Haren Publishing.

Tolido, R. (2009). Oracle OpenWorld: Innovation. 2009 style. | CTO Blog | Capgemini | Consulting, Technology, Outsourcing. Retrieved October 21, 2009, from http://www.capgemini.com/ctoblog/2009/10/oracle_openworld_innovation_20.php

Trapp, T. (2009). ABAP Software Ontologies. Retrieved September 30, 2009, from http://www.sdn.sap.com/irj/scn/weblogs?blog=/pub/wlg/15917%3Fpage%3Dlast%26x-maxdepth%3D0

Uschold, M. (2003). Where are the semantics in the Semantic web? AI Magazine, 24(3), 25–36.

Youseff, L., Butrico, M. & Da Silva, D. (2008). Toward a unified ontology of cloud computing. Paper presented at Grid Computing Environments Workshop, Austin, TX, November 16. Retrieved December 12, 2010 from http://freedomhui.com/wp-content/uploads/2010/03/CloudOntology.pdf

Zachman, J. & Sowa, J. (1992). Extending and formalizing the framework for information systems architecture. IBM System Journal, 31(3), 590-617.

Index

Acronyms

CG (or **CGs**): Conceptual Graph, Conceptual Graphs
CL: Common Logic
ISO: International Standards Organization
OpenSEA: Open Semantic Enterprise Architecture
TOGAF: The Open Group Enterprise Architecture Framework

Glossary

Conceptual Graphs (CGs) are a formal way of knowledge representation. Originally used to represent the conceptual schemas used in database systems, CGs have been applied a wide range of topics in artificial intelligence, computational intelligence, computer science, cognitive science and enterprise architectures.
ISO 24707:2007 Common Logic (CL) is a framework for a family of logic languages, based on first-order logic, intended to facilitate the exchange and transmission of knowledge in computer-based systems. The standard includes specifications for three dialects, the Common Logic Interchange Format (CLIF), the Conceptual Graph Interchange Format (CGIF), and an XML-based notation for Common Logic (XCL). Many other logic-based languages could also be defined as subsets of CL by means of similar translations; among them are the Semantic Web's RDF and OWL.
OpenSEA is a framework that combines the open semantics of TOGAF with the open syntax of ISO 24707:2007 Common Logic to provide an Open Semantic Enterprise Architecture.