

# An Approach to the Semantic Intelligence Cloud

Richard Greenwell

Submitted in Partial Fulfilment of  
the Requirements of Edinburgh Napier University  
for the Degree of  
Doctor of Philosophy

School of Computing

January 2018

## **Abstract**

Cloud computing is a disruptive technology that aims to provide a utility approach to computing, where users can obtain their required computing resources without investment in infrastructure, computing platforms or services. Cloud computing resources can be obtained from a number internal or external sources. The heterogeneity of cloud service provision makes comparison of services difficult, with further complexity being introduced by a number of provision approaches such as reserved purchase, on-demand provisioning and spot markets.

The aim of the research was to develop a semantic framework for cloud computing services which incorporated Cloud Service Agreements, requirements, pricing and Benefits Management.

The proposed approach sees the development of an integrated framework where Cloud Service Agreements describe the relationship between cloud service providers and cloud service users. Requirements are developed from agreements and can use the concepts, relationships and assertions provided as requirements. Pricing in turn is established from requirements. Benefits Management is pervasive across the semantic framework developed.

The methods used were to provide a comprehensive review of literature to establish a good theoretical basis for the research undertaken. Then problem solving ontology was developed that defined concepts and relationships for the proposed semantic framework. A number of case studies were used to populate the developed ontology with assertions. Reasoning was used to test the framework was correct.

The results produced were a proposed framework of concepts, relationships and assertions for a cloud service descriptions, which are presented as ontology in textual and graphical form. Several parts of the ontology were published on public ontology platforms and, in journal and conference papers.

The original contribution to knowledge is seen in the results produced. The proposed framework provides the foundations for development of a unified semantic framework for cloud computing service description and has been used by other researchers developing semantic cloud service description.

In the area of Cloud Service Agreements a full coverage of the documents described by major standards organisations have been encoded into the framework. Requirements have been modelled as a unique multilevel semantic representation. Pricing of cloud services has been developed using semantic description that can be mapped to requirements. The existing Benefits Management approach has been reimplemented using semantic description.

In conclusion a proposed framework has been developed that allows the semantic description of cloud computing services. This approach provides greater expression than simplistic frameworks that use mathematical formulas or models with simple relationships between concepts. The proposed framework is limited to a narrow area of service description and requires expansion to be viable in a commercial setting.

Further work sees the development of software toolsets based on the semantic description developed to realise a viable product for mapping high level cloud service requirements to low level cloud resources.

## **Acknowledgments**

I would like to thank my supervisors Prof. Xiaodong Liu and Dr. Kevin Chalmers and, my PhD panel chair Dr. Malcolm Rutter.

# **Publications**

## **Journal Articles**

R. Greenwell, X. Liu, and K. Chalmers, “Benefits Management of Cloud Computing Investments”, International Journal of Advanced Computer Science and Applications, vol. 5, no. 7, 2014.

R. Greenwell, X. Liu, and K. Chalmers, “An Upper Ontology for Benefits Management of Cloud Computing”, International Journal of Advanced Computer Science and Applications, vol. 5, no. 12, 2014.

## **Conference Papers**

R. Greenwell, X. Liu, and K. Chalmers, “Pricing Intelligence as a service for Cloud Computing”, IEEE CloudCom 2013, 2013.

R. Greenwell, X. Liu, and K. Chalmers, “Semantic description of cloud service agreements”, Science and Information Conference (SAI), 2015, pp. 823–831, 2015

R. Greenwell, X. Liu, K. Chalmers, and C. Pahl, “A Task Orientated Requirements Ontology for Cloud Computing Services”, 6th International Conference on Cloud Computing and Services Science (CLOSER), 2016.

R. Greenwell, X. Liu, K. Chalmers, “Pricing Ontology for Task-Oriented Cloud Sourcing”, 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud) p. 63, 2016.

# Table of Contents

<b>ABSTRACT .....</b>	<b>II</b>
<b>ACKNOWLEDGMENTS .....</b>	<b>IV</b>
<b>PUBLICATIONS .....</b>	<b>V</b>
<b>TABLE OF CONTENTS.....</b>	<b>VI</b>
<b>LIST OF FIGURES.....</b>	<b>XI</b>
<b>LIST OF TABLES .....</b>	<b>XIII</b>
<b>CHAPTER 1 INTRODUCTION.....</b>	<b>1</b>
1.1 Chapter Overview.....	1
1.2 Problem Statement.....	1
1.3 Aims and Objectives of the Research .....	5
1.3.1 Cloud Service Agreements .....	6
1.3.2 Requirements.....	6
1.3.3 Pricing .....	7
1.3.4 Benefits Management.....	7
1.4 Contributions to Knowledge .....	7
1.4.1 Contribution to Semantic Description of Cloud Service Agreements .....	9
1.4.2 Contribution to Cloud Service Requirements.....	9
1.4.3 Contribution to Cloud Service Pricing .....	11
1.4.4 Contribution to Benefits Management in Cloud Computing .....	12
1.5 Statement of Methodology .....	13
1.6 Criteria for Success.....	14
1.7 The Structure of the Thesis .....	15
<b>CHAPTER 2 LITERATURE REVIEW.....</b>	<b>17</b>
2.1 Chapter Overview.....	17
2.2 Cloud Computing.....	17
2.2.1 Introduction .....	17
2.2.2 General Concepts .....	19
2.2.3 Agreements for Cloud Computing .....	20
2.3 Customer Agreements in Cloud Computing .....	22
2.4 Service Level Agreements in Cloud Computing.....	26
2.5 Acceptable Use Policies in Cloud Computing.....	29
2.6 Requirements Engineering in Cloud Computing.....	29
2.7 Pricing of Cloud Computing Services .....	31
2.8 Business Aspects of Cloud Computing.....	32
2.9 Need for Representations .....	36
2.10 Description Logics.....	41
2.10.1 Introduction .....	41

2.10.2	Knowledge Reasoning Built on Description Logics .....	44
2.10.3	Applied Usage of Description Logics .....	46
2.10.4	Description Logics in Requirements Engineering .....	47
<b>2.11</b>	<b>Description Logics in Cloud Computing Service Specification.....</b>	<b>48</b>
2.11.1	Service Composition .....	48
2.11.2	Service Level Agreement Specification .....	49
2.11.3	Requirements Engineering .....	51
2.11.4	Security .....	51
<b>2.12</b>	<b>Problem Solving Semantic Description .....</b>	<b>52</b>
2.12.1	Origins of Problem Solving Ontology .....	54
2.12.2	Unified Problem-solving Method Development Language .....	55
<b>2.13</b>	<b>Discussion and Conclusions from Literature Review .....</b>	<b>56</b>
<b>CHAPTER 3</b>	<b>RELATED WORK .....</b>	<b>62</b>
<b>3.1</b>	<b>Chapter Overview .....</b>	<b>62</b>
<b>3.2</b>	<b>Introduction.....</b>	<b>62</b>
<b>3.3</b>	<b>Web 2.0 .....</b>	<b>63</b>
<b>3.4</b>	<b>Web 3.0 .....</b>	<b>64</b>
<b>3.5</b>	<b>Web 4.0 .....</b>	<b>66</b>
<b>3.6</b>	<b>Modelling the Semantic Web .....</b>	<b>67</b>
3.6.1	Semantic Mark up .....	69
3.6.2	Resource Description Framework (RDF) .....	71
3.6.3	Ontology Web Language (OWL) .....	73
3.6.4	JSON-LD .....	74
3.6.5	Triples Databases .....	75
<b>3.7</b>	<b>Semantic Rule and Query Languages .....</b>	<b>76</b>
3.7.1	RuleML .....	76
3.7.2	Ontology Rules.....	77
3.7.3	Querying Ontologies .....	78
<b>3.8</b>	<b>Semantic Web Services.....</b>	<b>79</b>
3.8.1	Web-Services Description Language .....	81
<b>3.9</b>	<b>Ontology Based Descriptions of Service Offerings.....</b>	<b>81</b>
3.9.1	Good Relations .....	82
3.9.2	Evidence Ontology (ECO) .....	82
<b>3.10</b>	<b>Technologies Strongly Related to Cloud Computing.....</b>	<b>83</b>
3.10.1	Edge and Fog Computing .....	83
3.10.2	Serverless and Lambda Computing .....	83
<b>3.11</b>	<b>Semantic Description Development and Toolsets .....</b>	<b>84</b>
3.11.1	Collaborative Ontology Development.....	84
3.11.2	Toolsets .....	86
<b>3.12</b>	<b>Discussion and Conclusions of Related Work .....</b>	<b>87</b>
<b>CHAPTER 4</b>	<b>OVERVIEW ON THE PROPOSED APPROACH .....</b>	<b>90</b>
<b>4.1</b>	<b>Chapter Overview .....</b>	<b>90</b>
<b>4.2</b>	<b>Introduction.....</b>	<b>90</b>
<b>4.3</b>	<b>Contribution of the Proposed Framework .....</b>	<b>92</b>
<b>4.4</b>	<b>Contribution to the Semantic Description of Cloud Service Agreements .....</b>	<b>92</b>

<b>4.5</b>	<b>Contribution to Cloud Service Requirements .....</b>	<b>94</b>
4.5.1	High Level Requirements.....	96
4.5.2	Low level Requirements.....	98
<b>4.6</b>	<b>Contribution to Cloud Service Pricing.....</b>	<b>98</b>
<b>4.7</b>	<b>Contribution to Benefits Management in Cloud Computing .....</b>	<b>99</b>
<b>4.8</b>	<b>Discussion and Conclusion .....</b>	<b>100</b>
<b>CHAPTER 5 SEMANTIC DESCRIPTION FOR CLOUD SERVICE AGREEMENTS .....</b>		<b>103</b>
<b>5.1</b>	<b>Chapter Overview.....</b>	<b>103</b>
<b>5.2</b>	<b>Introduction.....</b>	<b>103</b>
<b>5.3</b>	<b>Semantic Description for Cloud Service Agreements .....</b>	<b>105</b>
<b>5.4</b>	<b>Customer Agreements .....</b>	<b>105</b>
<b>5.5</b>	<b>Service Level Agreements .....</b>	<b>113</b>
<b>5.6</b>	<b>Acceptable Use Policy .....</b>	<b>117</b>
<b>5.7</b>	<b>CSA Case Study .....</b>	<b>119</b>
5.7.1	Purpose.....	119
5.7.2	System under Study.....	119
5.7.3	Application of the Proposed Method.....	120
5.7.4	Results .....	120
5.7.5	Evaluation.....	123
<b>5.8</b>	<b>Comparison to other Cloud Service Agreement Frameworks .....</b>	<b>124</b>
<b>5.9</b>	<b>Discussion and Conclusions .....</b>	<b>125</b>
<b>CHAPTER 6 REQUIREMENTS FOR SEMANTIC DESCRIPTION OF CLOUD COMPUTING SERVICES.....</b>		<b>127</b>
<b>6.1</b>	<b>Chapter Overview.....</b>	<b>127</b>
<b>6.2</b>	<b>Introduction.....</b>	<b>127</b>
<b>6.3</b>	<b>Requirements Semantic Description Design.....</b>	<b>127</b>
<b>6.4</b>	<b>Problem Solving .....</b>	<b>130</b>
<b>6.5</b>	<b>High Level Requirements - Brokerage.....</b>	<b>130</b>
6.5.1	Discovery .....	130
6.5.2	Mediation .....	131
6.5.3	Choreography .....	131
6.5.4	Adaptation .....	132
6.5.5	Grounding.....	133
6.5.6	Monitoring.....	133
6.5.7	Comparison .....	134
6.5.8	Fault Handling.....	134
6.5.9	Pricing (High Level).....	134
<b>6.6</b>	<b>Low Level Requirements.....</b>	<b>134</b>
<b>6.7</b>	<b>Specification .....</b>	<b>135</b>
<b>6.8</b>	<b>Requirements Case Study .....</b>	<b>137</b>
6.8.1	Purpose.....	137



6.8.2	System under Study .....	137
6.8.3	Application of the Proposed Method .....	138
6.8.4	High Level Problem-Solving Requirements Ontology .....	138
6.8.5	Brokerage .....	139
6.8.6	Discovery .....	139
6.8.7	Adaptation .....	139
6.8.8	Mediation .....	139
6.8.9	Grounding .....	140
6.8.10	Monitoring .....	140
6.8.11	Comparison .....	141
6.8.12	Fault Handling .....	141
6.8.13	Pricing (High Level) .....	141
6.8.14	Low Level Requirements Semantic Description .....	141
6.8.15	Mapping .....	141
6.8.16	Results .....	143
6.8.17	Evaluation .....	143
6.9	Comparison to other Cloud Computing Requirements Frameworks .....	143
6.10	Discussion and Conclusions .....	144
<b>CHAPTER 7 SEMANTIC DESCRIPTION OF CLOUD SERVICE PRICING ....</b>		
	.....	<b>146</b>
7.1	Chapter Overview .....	146
7.2	Introduction .....	146
7.3	Work flow design .....	148
7.4	Semantic Description of Pricing .....	153
7.5	Pricing Case Study .....	155
7.5.1	Purpose .....	155
7.5.2	System under Test .....	155
7.5.3	Application the of Proposed Method .....	155
7.5.4	Results .....	156
7.5.5	Evaluation .....	157
7.6	Comparison to Syntactical Pricing Frameworks .....	158
7.7	Discussion and Conclusions .....	160
<b>CHAPTER 8 SEMANTIC DESCRIPTION FOR BENEFITS MANAGEMENT OF CLOUD COMPUTING .....</b>		<b>161</b>
8.1	Chapter Overview .....	161
8.2	Introduction .....	161
8.3	Cloud Computing Enablers .....	164
8.4	Cloud Computing Case Studies .....	166
8.5	A Semantic Description for Benefits Management in Cloud Computing .....	176
8.6	Semantic Description Classes .....	177
8.7	Implementation .....	178
8.8	SPARQL Queries .....	179
8.9	Benefits Management Case Study .....	181
8.9.1	Purpose .....	181

8.9.2	System under Test .....	182
8.9.3	Proposed Method.....	182
8.9.4	Results .....	182
8.9.5	Evaluation.....	183
<b>8.10</b>	<b>Advantages of Reasoning .....</b>	<b>184</b>
<b>8.11</b>	<b>Comparison to other Business Benefits Frameworks for Cloud Computing.....</b>	<b>185</b>
<b>8.12</b>	<b>Discussion and Conclusions .....</b>	<b>186</b>
<b>CHAPTER 9</b>	<b>CONCLUSIONS AND FUTURE WORK .....</b>	<b>188</b>
<b>9.1</b>	<b>Chapter Overview.....</b>	<b>188</b>
<b>9.2</b>	<b>Introduction.....</b>	<b>188</b>
<b>9.3</b>	<b>Discussion of Results.....</b>	<b>189</b>
<b>9.4</b>	<b>Discussion of Evaluation.....</b>	<b>190</b>
<b>9.5</b>	<b>Critical Analysis.....</b>	<b>191</b>
9.5.1	General Effectiveness of the Semantic Framework.....	191
9.5.2	Cloud Service Agreements (CSA).....	193
9.5.3	Requirements.....	194
9.5.4	Pricing .....	194
9.5.5	Benefits Management.....	195
9.5.6	Conclusions .....	196
<b>9.6</b>	<b>Comparison to other Frameworks for Cloud Computing Service Description .....</b>	<b>197</b>
<b>9.7</b>	<b>Main Contributions and Conclusions .....</b>	<b>197</b>
9.7.1	Cloud Service Agreements.....	198
9.7.2	Requirements.....	199
9.7.3	Pricing .....	199
9.7.4	Advantages of a Semantic Description Framework for Benefits Management .....	200
9.7.5	Conclusions .....	201
<b>9.8</b>	<b>Future Work.....</b>	<b>203</b>
9.8.1	Expansion of Semantic Framework via the Web Protégé Ontology .....	203
9.8.2	Usage of Semantic Framework by Organisations .....	203
9.8.3	Cloud Service Agreements .....	204
9.8.4	Semantic Description for Requirements.....	205
9.8.5	Pricing .....	205
9.8.6	Benefits Management.....	205
9.8.7	Towards a Unified Semantic Framework for Cloud Service Description .....	206
<b>REFERENCES</b>	<b>.....</b>	<b>207</b>
<b>APPENDIX A - ABBREVIATIONS AND ACRONYMS .....</b>		<b>226</b>
<b>APPENDIX B – UPML ONTOLOGY .....</b>		<b>228</b>
<b>APPENDIX C – AN UPPER ONTOLOGY FOR BENEFITS MANAGEMENT OF CLOUD COMPUTING INVESTMENTS .....</b>		<b>230</b>
<b>APPENDIX D – HYBRID META-HEURISTIC PRICING ONTOLOGY .....</b>		<b>231</b>

## List of Figures

Figure 1 - Atomic Concepts .....	42
Figure 2 - Atomic Roles .....	42
Figure 3 - Problem Solving Ontology .....	52
Figure 4 - Microdata Example .....	70
Figure 5 - Microformat Example .....	70
Figure 6 - RDF Triple.....	71
Figure 7 - RDF Ontology Fragment .....	72
Figure 8 - Example of JSON-LD .....	75
Figure 9 - Example of RuleML .....	77
Figure 10 - Example of a SWRL Rule .....	78
Figure 11 - Example SPARQL Query .....	78
Figure 12 - Protege Ontology Editor .....	87
Figure 13 - Overview of Semantic Framework .....	91
Figure 14 - Cloud Security Concepts.....	104
Figure 15 - RBox Axioms for Security .....	104
Figure 16 - ABox for Security .....	105
Figure 17 - Usage and Payment Structure TBox.....	107
Figure 18 - Usage and Payment ABox .....	107
Figure 19 - Temporary Suspension of Agreement TBox .....	108
Figure 20 - Temporary Suspension ABox .....	108
Figure 21 - Legal Aspects of CSA Semantic Description .....	109
Figure 22 - Legal Aspects of CSA ABox.....	110
Figure 23 - Security TBox.....	110
Figure 24 - Risk TBox.....	111
Figure 25 - Example Risk Score Calculation .....	112
Figure 26 - Risk Probability ABox.....	112
Figure 27 - SLA Terminology and Relations.....	113
Figure 28 - Quality of Services Terminology and Relations.....	114
Figure 29 - Data Aspects Terminology and Relations .....	115
Figure 30 - Rights Terminology and Relations .....	116
Figure 31 - Liabilities Terminology and Relations.....	116
Figure 32 - Acceptable Use Policy TBox Overview .....	117
Figure 33 - Terminology surrounding AUP Terms .....	118
Figure 34 - Acceptable Use Policy ABox.....	118
Figure 35 - Acceptable Use Policy RBox.....	119
Figure 36 - Oracle Payment Structure ABox .....	120
Figure 37 - Oracle Temporary Suspension ABox .....	121
Figure 38 - Oracle Terms and Conditions Abox .....	121
Figure 39 - Oracle Security and Risk ABox .....	121
Figure 40 - Oracle Acceptable Use Policy ABox .....	121
Figure 41 - IBM Payment Structure ABox .....	122
Figure 42 - IBM Temporary Suspension ABox .....	122
Figure 43 - IBM Terms and Conditions ABox .....	122
Figure 44 - IBM Security and Risk ABox .....	123

Figure 45 - IBM Acceptable Use Policy ABox.....	123
Figure 46 - Requirements Semantic Description Implementation.....	128
Figure 47 - The Hierarchical Structure of the Semantic Description.....	129
Figure 48 - Discovery Element Semantic Description Fragment .....	136
Figure 49 - Electromagnetism (EM) Meta-Heuristic.....	150
Figure 50 - Workflow for Pricing Cloud Computing Requirements .....	152
Figure 51 - High Level Problem Solving Ontology for Pricing.....	153
Figure 52 - Trigger service requests to cloud API .....	154
Figure 53 - Pricing Tasks ABox Entries.....	156
Figure 54 - Pricing Resource Usage ABox.....	156
Figure 55 - IBM Payment Structure ABox .....	157
Figure 56 - Benefits Dependency Network.....	163
Figure 57 - Business Enablers .....	170
Figure 58 - Operational Enablers .....	172
Figure 59 - Overview of Main Semantic Description Concepts.....	177
Figure 60 - Overview of Main Semantic Description Concepts.....	178
Figure 61 - Semantic Framework Implementation for Benefits Management.	179
Figure 62 - Benefits Management Abox Encoding .....	183

## List of Tables

Table 1 - Customer Agreements (CA) .....	23
Table 2 - Service Level Agreement (SLA) .....	26
Table 3 - Acceptable Use Policy Terms (AUP).....	29
Table 4 - Dissatisfaction levels with benefits derived from IS/IT activities .....	34
Table 5 - Description Logic Constructors .....	43
Table 6 - Risk Probability Score Calculation .....	112
Table 7 - Discover Resources Task .....	136
Table 8 - Case Study: High Level Requirements.....	138
Table 9 - Pricing Differentials Based on Geographic Location .....	148
Table 10 - CPU Requirements for EM HMH.....	151
Table 11 - Pricing for Processing Power Taken from Yeo et al. ....	158
Table 12 - Organisations Reviewed .....	167
Table 13 - Enablers cross-referenced to case studies .....	168
Table 14 - Classification of Benefits .....	175
Table 15 - Cloud Investment Portfolio .....	176
Table 16 - Benefits Satisfied by SPARQL Queries .....	181

# **Chapter 1      Introduction**

## **1.1 Chapter Overview**

This chapter provides an introduction to the problem statement, which views cloud computing services as a complex set of resources that are currently described and managed using simplistic approaches.

The four aspects of the proposed framework (agreements, requirements, pricing and benefits management) are introduced, along with the associated contributions to knowledge.

The statement of methodology and criteria for success are established. The structure of the thesis is then defined.

## **1.2 Problem Statement**

Cloud Computing services comprise a complex set of resources that are organised to provide a service offering. The service offering is delivered at a defined service level defined by an agreement and, at a price over a timeframe or at a usage level. A major feature of cloud computing services is customer self-service, which is the ability of a customer to build service offerings from a set of resources from a notional markets for a time or usage level.

Attempts have been made to express service levels and pricing of cloud services using simplistic 'syntactical' approaches'. Cloud services and inter-relationships between cloud services have been described as simple contracts, in terms of mathematical variables and using non-descriptive relationships, an example being Amazon EC2 where [1] where services are described in terms of price/usage terms. This leads to a number of issues:

- Information is lost, as the modelling simplicity fails to capture the true complexity of service offerings.
- The service user is unable to consider all service offerings in a rational manner, as their service selection decisions may be incorrect.
- Prices paid for services may be too high or the benefits derived from service usage may be under or over-estimated

In this thesis an attempt has been made to develop a new unique approach to modelling cloud services and composition using semantic modelling approaches and not only consider service level aspects of cloud services, but also pricing and economic and other benefits which have not been considered by existing research.

The research question is “Can Cloud Service Agreements, requirements for cloud services, pricing of cloud services and benefits derived from cloud services be modelled using semantic techniques, to aid customer self-service of cloud resources?”

Cloud service usage and composition research has concentrated the issues shown in the list below:

- i. Low level infrastructure provision, such as description of virtual machines as a set of resources such as CPU, Memory and Storage.
- ii. Applications or Services that can be created or represented as cloud services. Software as a Service (SaaS) and Platform as a Service (PaaS) and applications such as MapReduce (for example Hadoop) which have been viewed as suited to running on cloud services.
- iii. Service Level Agreements (SLA) for minimum service vendor performance, for example service response time or minimum service provision.
- iv. Legal and contractual aspects of cloud services
- v. Edge and Fog computing

(i) Low level infrastructure provision

Low level infrastructure provision is concerned with the supply of virtual hardware and operating system level resources as virtual machines. Cloud providers will supply virtual machines with characteristics such as CPU, memory, hard-disk and operating systems, for a price for a rental period, platforms which supply infrastructure and software development resources or services which supply 'rental' of a software functionality.

A major aspect of infrastructure provision is customer self-service. Customers can select from a large number of combinations of machine characteristics, there are millions of combinations of machine choices and it is difficult for customers to make selections and consider the impact of selection on price. The current service offerings are specified in terms of simple syntactical descriptions, such as spreadsheets or websites that present the combinations of machine characteristics to derive a price. It is difficult to see deeper semantic relationships between machine characteristics, for example, is the provision of processing resources from one supplier the same as the provision from another supplier?

Monitoring and in-situ adjustment of virtual machines is another aspect of service description that require a deeper meaning to be defined between static service description when virtual machines are first specified and when they are deployed and running.

(ii) Cloud based Applications and Services

Software applications can be delivered as services, as with infrastructure the services have characteristics, such as number of user licences or storage space at a given price for a rental period. Platforms that are specified above and abstract low level infrastructure are specified in terms of Application Programmer Interfaces (API).



Resources such as scientific calculation services also have characteristics as number of calculations that can be carried out in a given time for a given price. Emerging technologies such as lambda clouds [2]. Lambda clouds are cloud computing resources that allow individual calculations to be specified and combined using lambda calculus. Lambda calculus is a form of calculus which allows problems to be specified as functions which are bound to form calculation systems

### (iii) Service Level Agreements

A major research area of low level provision of virtual machines and of higher level services has been Service Level Agreements (SLA). A SLA has typically been specified by cloud service providers to describe how a service should perform in terms of availability, reliability and minimum performance characteristics. Many vendors will provide services over geographic regions which will have different characteristics, such as network latency. Many SLA have been defined in simplistic syntactical forms, which make it difficult to compare or match services in 'markets' or simulations of markets. An example being the VieSLAF model [3] which provides simplistic word and structure mapping of SLA expressed as XML documents.

Closely related to SLA are Customer Agreements (CA), which concentrate on the initial agreement between a service provider and customer, and Acceptable Use Policies (AUP), which describe how each side should act after an agreement is made.

### (iv) Legal and Contractual Aspects of Cloud Services

CA and AUP are more formal contractual/legal aspects of cloud service description and procurement. The legal aspects of cloud services centre on contract law in a number of jurisdictions. The main legal systems being Common Law, based on previous law cases and Civil Law based on rules or

codifications, with some jurisdictions based on a hybrid of Common and Civil Law.

A cloud service description may cross a number of jurisdictions and the deep meaning and implications of contracts needs to be considered in terms of semantic meaning as differently defined, but semantically similar concepts must be matched and compared.

(v) Edge and Fog computing

Edge and Fog computing extend cloud computing to combine computing geographically dispersed location aware Internet of Things (IoT) devices and cloud computing services. Devices can be seen as resources extending cloud services. The characteristics of the devices on the edge of the network, can be viewed as a number of nodes, which are location aware and widely geographically distributed. There is interaction between the cloud and the fog/edge services, with cloud computing powerful compute intensive services, for example machine learning and data science services and a device such as mobile phone providing user interaction services. The need for complex semantic description becomes even greater, given the increase in the number of components, relationships and interdependencies.

### **1.3 Aims and Objectives of the Research**

This sub-section describes the aims and objectives of the research undertaken. A proposed semantic framework for cloud computing services has been developed for description of aspects cloud computing services in the following areas, shown in the list below:

- Cloud Service Agreements
- Requirements
- Pricing
- Benefits Management

### **1.3.1 Cloud Service Agreements**

Cloud Service Agreements (CSA) describe the relationship between cloud service providers and cloud service users.

The aim of the research undertaken was to develop a semantic description of CSA, so that agreements from a number of providers can be compared. It was found that CSA comprise, shown in the list below::

- Customer Agreements
- Service Level Agreements
- Acceptable Use Policy

The objectives of the research were to develop semantic description for the three areas described above and, to highlight common elements in the semantic description of the areas.

### **1.3.2 Requirements**

Requirements for cloud services are developed from Cloud Service Agreements.

The aims for research into cloud service requirements were to explore the use of Problem Solving Ontology (PSO) to describe high level requirements supplied by a cloud service consumer, which are mapped to low level requirements which map the high level requirements to cloud service resources.

The objectives of the research were to describe the areas in the list below:

- Problem Solving Ontology
- Each aspect of high level requirements as PSO
- Each aspect of low level requirements as PSO
- Mappings between high and low level requirements

### **1.3.3 Pricing**

Pricing of cloud services is based on requirements, with pricing seen in both high and low level requirements. The aim of the research in this area was to provide a detailed examination of semantic description of cloud service pricing.

The objectives of the research into cloud service pricing were firstly, to provide a framework for pricing that forms part of proposed framework for semantic description of cloud computing services. A second objective efficacy of the pricing framework was demonstrated by encoding a pricing case study using the proposed framework.

### **1.3.4 Benefits Management**

Benefits Management is seen as pervasive in the proposed framework for description of cloud computing services.

The aim of the research was to show how benefits from cloud computing services could be described semantically.

The objectives of the research were to take the existing Benefits Management approach [4] and enhance the approach using semantic techniques. The enhanced approach was used to encode a number of case studies. Results from encoding were used to fulfil the objective of identifying common benefits expected from cloud computing services.

## **1.4 Contributions to Knowledge**

The main contribution to knowledge is to propose a semantic framework for description of cloud computing services. The proposed framework brings together abstract semantic models to cloud infrastructure and service descriptions and to provide concrete examples of how cloud resources could be modelled to introduce semantic concepts.

The framework is built on semantic descriptions and a specialism of a semantic description Problem Solving Ontology (PSO) which uniquely models cloud computing requirements as tasks which are solved by generic Problem Solving Methods (PSM) which work against a number of knowledge domains. Given a common representation it is possible to map and trace semantically equivalent elements from agreements, through requirements and pricing, with the benefits from cloud computing services being assessed throughout the service description process. The framework is seen as the first phase in a “Unified Semantic Framework for Cloud Service Description”

A number of unique studies for Cloud Service Agreements, requirements, pricing and benefits derived from cloud resources have been carried out. The studies draw upon some techniques applied to other fields of computing and economics, which are uniquely applied to cloud computing. These areas have not been widely covered by other researchers and, in particular the models of benefits management research, which is seen as unique in cloud computing research. These areas of research are highly important to the key aim of user self-service.

The semantic representation of cloud services described in the proposed framework provides an illustration which can be applied to the ‘wider cloud’.

Specific contributions to knowledge provided by the proposed framework are identified in four areas, shown in the list below:

- Contribution to Semantic Description of Cloud Service Agreements
- Contribution to Cloud Service Requirements
- Contribution to Cloud Service Pricing
- Contribution to Benefits Management in Cloud Computing

Further work in section 9.8.7 sees the proposed framework being developed further into ‘Unified Semantic Framework for Cloud Computing Services’ where the four areas described above and the relationships between the

elements are merged into a single semantic description of cloud computing services.

#### **1.4.1 Contribution to Semantic Description of Cloud Service Agreements**

A majority of research into Cloud Service Agreements (CSA) has concentrated on “syntactical” description of mainly Service Level Agreements (SLA), that is, the description of SLA without separation of terminology from relationships and assertions, using simplistic descriptions.

Using an extensive literature review [5] [6] it was found CSA comprised not only SLA but also Customer Agreements (CA) (agreements between cloud service providers and their customers) and Acceptable Use Policies (AUP). A major contribution is to provide a unique semantic description of these artifacts and their interactions. This also led to consideration of further semantic description shown in the list below:

- Requirements
- Pricing
- Interaction with semantic description of legal and contractual aspects of cloud services developed by other researchers

#### **1.4.2 Contribution to Cloud Service Requirements**

Unique semantic description of terminology, relationships and assertions were developed by describing cloud service requirements as problem solving semantic framework, which provides description of activities as tasks, generic problem solving methods and knowledge domains. Two additional layers of the semantic framework were developed by ‘overlying’ concepts as a brokerage or high level layer which analysed Cloud Service Agreements and mapped them to concepts identified in the brokerage process, listed below:

- Discovery – Requirements for finding suitable service or components
- Mediation – Requirements to resolve differences between user requests and services offered by cloud service providers
- Choreography – Requirements for organising cloud services or components
- Adaption – Requirements for adjusting or altering services or components
- Grounding – Requirements for communication between high level requirements, low level requirements and the cloud service provider
- Monitoring – How the service user will monitor the cloud service they are using
- Comparison – Requirements of how cloud services can be compared semantically
- Fault Handling – Requirements for handling malfunctions in cloud services
- Pricing – How cloud services will be priced

Brokerage or high level requirements were mapped to low level requirements. Low level requirements map onto physical cloud services. Low level requirements are described in the list below:

- Resource Description – Description of available cloud services as semantic description, independent from service suppliers.
- Pricing – Pricing information supplied by cloud service providers using semantic description, so that it is independent from service suppliers. So that customers can compare the prices of equivalent services.
- Cloud Interfaces Adapters and Bridge – Requirements describing how high level requirements will be mapped to physical cloud services, for example how an algorithm will be mapped to CPU and memory requirements.

The semantic description developed provided a number of benefits, described in the list below:

- The ability to map onto CSA.
- The identification of key high level\brokerage elements for cloud services.
- The ability to map high level requirements to low level requirements and, to trace the origins of low level requirements to high level requirements.
- The ability to map requirements to benefits specified as key business drivers for cloud services.
- The identification of pricing as a key factor in high level and low level requirements.
- The ability to feedback and feedforward requirements from/to cloud services and requirements.

A unique case study was developed that demonstrated the efficacy on the unique Problem Solving Ontology (PSO) based requirements semantic description.

### **1.4.3 Contribution to Cloud Service Pricing**

Utilising the work into the semantic description of cloud service requirements, which identified pricing as area unique to cloud service description, when compared to previous semantic service descriptions such as semantic web services. A semantic description was developed for pricing of cloud services.

Many researchers have concentrated on “syntactical” or mathematical models of cloud service pricing. The contribution provided in this area is to make available terminology and relationships for cloud service pricing, giving the benefits of abstraction from individual cloud service implementations and public cloud service markets. New knowledge can be created reasoning



across the semantic descriptions generated, for example via inheritance inferred from semantic relationships.

#### **1.4.4 Contribution to Benefits Management in Cloud Computing**

Outwith broad economic and generic business research there has been little research into the benefits generated by cloud computing services. These benefits are closely related to work into pricing, requirements and Cloud Service Agreements. Pricing is associated to cost/benefit decisions taken when utilising a cloud service. Requirements must be linked to benefits derived from requirements. Cloud Service Agreements must also link to benefits a customer expects to derive from a cloud service.

A major contribution to knowledge was generated by applying an existing benefits management approach to cloud computing services, which had previously been applied to non-cloud architectures. This gave a two-fold contribution, shown in the list below:

- Codification of the benefits management approaches concepts and relationships as semantic description
- Codification of the author's previous research case studies and 3<sup>rd</sup> party research case studies into semantic description assertions (knowledge from the case studies expressed in terms of the concepts and relationships)

The codification of the benefits management approach allows other researchers and practitioners to develop their own assertions gathered from research to express benefits management work as semantic description.

The codification of case studies research provides a threefold contribution to knowledge. Firstly, a methodology and toolset for researchers and practitioners to codify their own work. Secondly, the ability search and reason across the case studies already codified and additional case studies,

to understand existing knowledge or develop new knowledge. Lastly, to develop benefits management concepts, relationships and assertions as collaborative semantic description and ontology.

## **1.5 Statement of Methodology**

The methodology used is one of synthesis identified by Cooper [7] as the connecting multiple research sources and summarising them to create the basis for new and novel research. This uses the scientific reductionist approach [8] which aims to analyse complex systems to extract simplified behaviour from complex relationships and interactions.

A number of semantic modelling techniques were considered and one has been selected. The current approaches to describing cloud resources and agreements are analysed and the essential models and data are extracted to form new models which are examined and criticised. Models and frameworks seen in other areas of computing are applied to cloud computing to assess their usefulness.

Emerging technologies are examined and the techniques synthesised from previous work is developed into possible future work.

The aim of the methodology is to synthesise a semantic modelling framework that is applicable to description of cloud resources. The framework allows researchers and users to consider cloud agreements, requirements, pricing and benefits management of cloud computing services. The framework was synthesised from a number of primary and secondary research sources, to develop a semantic representation that describes cloud services and the interaction between aspects of cloud services.

## 1.6 Criteria for Success

In addressing the research question a number of semantic representation techniques have to be considered. The representation techniques must be capable of modelling a number of aspects of cloud computing (infrastructure, services and agreements).

The criteria for success are described in the list below:

1. Selection of an appropriate semantic representation amenable to machine representation
2. Application to cloud service scenarios
3. Development of working semantic models

The first criteria for success is the ability to select an appropriate representation that can allow sufficient semantic expression, which is amenable to machine representation and processing. An emerging criteria for success is the ability to produce information such as pricing information and possible benefits from cloud resources sufficiently quickly to allow users to make decisions in fast moving markets.

The second criteria for success to apply semantic techniques to cloud service scenarios that to produce superior performance when compared with current syntactical representations. Scenarios are gathered from academic literature and remodelled using semantic techniques.

Development of working semantic models for a number of aspects of cloud computing such as service agreements, requirements, pricing and benefits managements will demonstrate that the semantic techniques provide value in modelling cloud resources is seen as a third success criteria.

## **1.7 The Structure of the Thesis**

The thesis structure is now briefly outlined. Chapter one provides an overview of the research, outlining the problem statement including a research question. This is followed by the aims and objectives of the research and the contributions to knowledge. The statement of methodology describes the synthesised approach followed in this thesis. The three criteria for success are then proposed.

The literature review is presented in Chapter two. Section two of the literature review is concerned with an introduction to cloud computing and presents a number of cloud computing concepts such as cloud service agreements, requirements for cloud service description, pricing of cloud services and benefits management of cloud service investments. There is a brief overview of requirements engineering in cloud computing. Sections three to eight examine semantic approaches to knowledge representation that may be useful for describing cloud computing resources. The state of the art in cloud service specification is dealt with in Sections nine to twelve. Section thirteen provides a discussion of the literature review.

Chapter three provides an overview of related work, with Sections three to five outlining technologies seen in the “semantic web”. Problem solving semantic description, which is a specialised semantic representation for problem based scenarios is discussed in the second section. The sixth to eleventh sections examines semantic web services, which is followed by practical semantic description frameworks and ontologies already used in academia and industry.

The fourth chapter describes an overview of the proposed approach and the contribution to research that has been made. The methodology followed, which aims to expand the statement of methodology and how the research will meet the criteria for success to answer the research question.

Semantic description models for cloud resources are provided in chapters five, six, seven and eight. These are key deliverables from the research undertaken.

Chapter five focusses on Cloud Service Agreements. These agreements control the interactions between cloud service users and cloud service providers. A unique contribution has been made by describing the three artifacts which make up Cloud Service Agreements, which are Customer Agreements, Service Level Agreements and Acceptable Use Policies in a semantic framework. Customer Agreements concentrate on the requirements of cloud service users. Service Level Agreements describe the standards of services supplied and penalties for missing agreements. Acceptable Use Policies designate user actions that may cause breaches of acceptable behaviour and sanctions for such behaviour.

The sixth chapter examines requirements for cloud service expressed in a two level semantic framework. High level requirements focus user and brokerage requirements for cloud services. Low level requirements are mapped to high level requirements and provide a layer of abstraction of the interface to physical cloud services.

Chapter seven provides an examination of cloud service pricing. Pricing is a unique feature of cloud computing services, when compared to previous technologies, such as semantic web-services. A semantic description was developed to abstract pricing descriptions from cloud service providers.

Benefits management of cloud computing investments was investigated in chapter eight. An existing framework which was “syntactical” in nature, was transformed into a semantic framework. The new framework was used to analyse a number of case studies from primary and secondary sources. An ontology was developed from the semantic analysis.

The thesis is completed with conclusions and, future work in Chapter nine.

## **Chapter 2      Literature Review**

### **2.1 Chapter Overview**

An extensive review of literature is undertaken in this chapter. The general concepts of cloud computing are introduced.

A detailed literature review of the four aspects of the proposed framework (agreements, requirements, pricing and benefits management) is carried out.

The review then moves on to the need for cloud service description representations and, then examines possible semantic representations that could be suitable.

### **2.2 Cloud Computing**

#### **2.2.1 Introduction**

Many organisations are considering cloud computing as a major aspect of their information systems strategy and, have made significant investments in cloud technology. The increasing maturity and uptake cloud computing will require organisations to consider the business and organisational value they gain from such investments. Cloud computing has been compared to the time sharing computer services that were prevalent in IS systems in the 1960's and 1970's [9]. Organisations have attempted to outsource non-core business activities to specialist providers, when cost effective solutions can be found.

Grossman [10] defines cloud computing as “clouds, or clusters of distributed computers, providing on-demand resources and services over a network, usually the Internet, with the scale and reliability of a data center”.

Organisations can use combinations of hardware and software as required to deliver services. Provision of information systems are outsourced rather than maintaining in-house infrastructure, information systems development platforms and applications software services. It is important to consider how organisations use these provisions and manage ownership models presented in cloud computing.

A number of provision models for cloud computing exist, shown in the list below:

- Infrastructure as a Service (IaaS) is a basic level of generic hardware support.
- Platform as a Service (PaaS) builds on IaaS and brings together infrastructure, operating systems, programming languages and data storage services.
- Software as a Service (SaaS) builds on PaaS and provides the ability to ‘rent’ software for periods of time for a selected number of users.

The models have common features, shown in the list below:

- Rental model of ownership
- Elasticity of service usage
- Flexibility of information storage and user self-service

The differences in the models also affect the view of cloud computing, IaaS can be seen as utility services and PaaS and some SaaS represent portfolios of business services that can be used in business transformation.

A number of cloud ownership models have been observed, shown in the list below:

- Public clouds are provided by a third party and are rented by customers
- Private clouds are created and maintained by a single entity
- Hybrid clouds use a combination of public and private clouds

Cloud computing can be linked to a number of strategic innovations such as Big Data [11] and Data Science [12]. Low cost ubiquitous cloud computing resources can be used to process large amounts of information from large datasets or databases (Big Data) and complex statistical and machine learning techniques can be applied to data sets (Data Science). These activities were previously carried out by large organisations with expensive bespoke information systems, such as grid or super computers.

### **2.2.2 General Concepts**

The National Institute of Standards and Technology (NIST) [13] provides a general definition of cloud computing. Firstly, an on-demand self-service. A user can select cloud services such as compute or storage resources without the need to interact with human suppliers. Secondly, the ability to access services through a number heterogeneous mechanisms, via a number of clients. Resource pooling sees the ability of cloud service providers to provide virtual resources to customers without disruption or cognisance of service delivery mechanisms, users can demand additional resources or free resources seamlessly from the resource pool. The ability to utilise the resource pool is seen in the concept of rapid elasticity, the users can choose to expand or reduce their resource usage at will in line with some contract or service level agreement. Finally, the service can be measured by some abstract methodology such as virtualised processing



usage, storage level or bandwidth usage, this will in turn be translated into a “price” or measure of utility.

Gubbi et al. [14] describe the Internet of Things (IOT) as being as ubiquitous intercommunicating wireless devices organised in a network. Cloud computing and IOT are being combined into single solutions, with large processing tasks being carried out on the cloud and the results being sent back to devices. This furthers the need for sophisticated service description, agreements, composition and pricing.

Mobile computing is becoming increasingly important in cloud computing. Sharma et al. [15] see mobile cloud computing as moving computing power and storage away from the mobile computing device to the cloud. The driving force behind fog or edge computing is to bring computing power to close to the ‘edge’ of the network to provide lower latency to mobile device users. Datta et al. [16] see fog computing as the major enabler for the IOT for applications such as connected vehicles, to provide the low network latency, delivered by road side units and machine to machine (M2M) communication.

NIST [13] describes a ‘traditional view of cloud services as infrastructure, platform or service. There is now a move to ‘sub-infrastructure’ services such as Lambda Cloud. Jonas et al [17] describe Lambda clouds providing functionality as stateless functional services, or ‘serverless’ services. This provides the ability for vendors to provide services at a higher granularity.

### **2.2.3 Agreements for Cloud Computing**

A utility market for cloud computing has been compared to a 5<sup>th</sup> utility [18] such as electricity or water. To establish a true utility market, user requirements should be submitted to a marketplace and, the requirements are then mapped to available service resources at an agreed price.

NIST [19] identifies the need to establish cloud service metrics, including standardised units of measurement for cloud resources. There are no common collections of vendor agreed terms. Storage and access to storage over a network vary. Service providers have not defined and applied standardised units of measurement.

CSCC [6] defines “Three Artifacts” of Cloud Service Agreements (CSA) as Customer Agreements (CA), Service Level Agreements (SLA) and , Acceptable Use Policies (AUP). SLA have been discussed by many researchers using simplistic mathematical or “syntactical” models (simple non-semantic models). It is important to provide an overview of SLA in the introduction and in the discussion of related work, as there is overlap and integration with the two other artifacts seen in CSA.

The Cloud Standards Customer Council (CSCC) [6] outline a number of problem areas, such as lack of standard nomenclature for CSA terms, lack of care in the drafting of some CSA and the poor of precision in semantics of agreements, that can radically alter their meaning.

CSCC [6] goes on to outline the requirements of CSA as having a number of intrinsic properties such as clarity, brevity, completeness, focus and changeability. There are also extrinsic properties of CSA such as comparability and understandability.

To provide cloud services as a utility there must be a mechanism for information interchange, measurement and agreement between consumers and service providers. Sheth and Ranabahu [20] discuss interoperability of cloud services as being a major challenge. The user has to select cloud services dependent on their application requirements, cost constraints, legal constraints and, the level of service required. Requirements and constraints must be mapped to a service provider’s architecture. A software service is developed or configured using the requirements and constraints. Problems

arise if users have to change providers and need to rewrite or reconfigure applications and data.

Ward et al. [21] describe a model for outsourcing contracts which could be applied to CSA. The model is comprehensive and deals with a detailed breakdown of most aspects of CSA, such as the parties involved in the contract, the contents of the services provided, pricing and penalty clauses and measurement criteria. Although a simplistic (syntactical) model is presented in UML, there is lack of a formal semantic model with no delineation between terminology, relationships and assertions.

A number of researchers have used syntactical approaches to describe CSA, however, the approaches are not capable of modelling the complex relationships between parties. A semantic approach is required to model relationships and to map concepts between the aspects of CSA.

Researchers have concentrated on specific domains of investigation such as SLA and legal agreements or pricing. Few researchers have examined customer agreements and acceptable use policy in a holistic manner. The common aspects between the cloud agreement knowledge domains also requires further investigation.

## **2.3 Customer Agreements in Cloud Computing**

Customer Agreements (CA) define how customers will use the service offerings. Customer Agreements (CA) have been investigated by a number of researchers. Usage of cloud services has been investigated by Greenwell et al. [22] using the Benefits Management approach developed by Ward and Daniel [4]. A number of case studies were used to examine how cloud technology enables changes within organizations which in turn create business benefits. It was found that many larger organizations perceive cloud services as utilities with price as a main driver. Smaller organizations can generate competitive advantage by using unique cloud characteristics,

such as the ability to access compute and storage capacity on demand at low cost.

The characteristics of the customer agreement are defined in Table 1 below.

Characteristic	Description
Usage	How the user is expected to utilize the cloud service.
Fee and payment	How much the user expects the service to cost?
Temporary Suspension	Factors that trigger the temporary suspension of cloud services.
Legal Terms and Termination	The terms and conditions within the agreement and actions that would cause the termination of the service agreement.
Indemnification Disclaimer	Claims, damages and losses not attributable to the supplier.
Limitation of Liability	Limits on obligations of parties in a contract
Security and Privacy	Security of data provided by the customer and supplier and data protection issues.

**Table 1 - Customer Agreements (CA)**

Taking the first characteristic of usage. Usage or intention to use is based on the perception of users towards cloud services [23], usefulness and trust are seen as key characteristics. Perceived usefulness defines the characteristics that user expects to enhance their performance or increase their utility.

Perceived ease of use relates to how easy it is for users to access and utilise features of the cloud service, this relates strongly to the self-service aspects of cloud service provision. Butt et al. [24] identify Cloud Self Service (CSS) as a major driver in the uptake of cloud computing services, allowing users to define and utilise cloud resources without recourse to system or

database administrators. Cloud service providers must embed technical provision capabilities into the cloud service.

Ubiquity of cloud services allows users perceive an unbroken or continuous access to cloud services. This increases user's propensity to use cloud services, as they will have to make minimum effort to start using a cloud service that is always available.

Trust in cloud services is based on security and privacy aspects of cloud services. This is discussed in more detail below as these are specific aspects of cloud computing identified by the CSCC.

Fee and payment aspects of cloud services have concentrated on market/contract models. A typical example of the market based approach is seen in Menychtas et al. [25]. The authors describe an information stage where the cloud offerings are described, a negotiation phase where an agreement is made between the cloud service provider and consumer, a contracting phase where the agreement is finalised and a settlement phase when consideration (payment) for the service is made. A number of payment strategies are advanced by researchers. Cao et al. [26] describe payments for reserved instances of cloud resources made for a discount and more expensive on demand instances made on a pay-as-you-go basis (on demand). More recently spot markets have developed which have variable pricing as cloud service providers attempt to sell unused capacity at a discount [27]. Cloud users must build a portfolio of cloud resources from reserved, on-demand and spot resources driven by their requirements over time.

The reasons for temporary suspension of customer agreement will be defined in the CA. The reasons for suspension may well be the misuse of a cloud service, for example use of a cloud service to launch a denial of a service attack or to store illegal content. Temporary Suspension involves the

customer's services being withdrawn for a number of reasons, such as abnormal usage of cloud resources, security risks and late payments [6].

The legal aspects of CA comprise the terms and conditions including termination clauses and limitations on liability (Indemnification, disclaimers and limitation). Bradshaw et al. [5] describe a Terms of Service (ToS) document, that forms part of customer agreement. The ToS will contain legal clauses such as choice of law, for example UK law, contract duration and renewal period and fee structure. Indemnification relates to freedom from liability for losses from that a customer may incur from loss of service (unscheduled downtime) or security or privacy breaches [28].

Security and Privacy characteristics of CA are described in Xiao and Xiao [29]. The researchers describe confidentiality, integrity, availability, accountability and privacy. Confidentiality involves preventing data access from the cloud provider, other customers and from external access. Pearson [30] describes the privacy aspects of cloud computing which form part of a CA. Data must be collected legally and must be up-to-date. Data collection must have a purpose and, its usage must be limited. Data should be kept securely and, there should be accountability for its usage. Knowledge of data collection and storage should be published openly. Individuals should be able to challenge data usage.

Integrity of both cloud processing and storage should be guaranteed by the cloud service. Integrity implies there is no corruption in processing results or data stored on the cloud storage.

Cloud processing and storage should have high levels of availability, this is seen as a core part of the SLA. Without high availability the cloud service offering would be unviable.

The ability to monitor changes in cloud resources to a low level (virtual machine or virtual storage levels) is covered by accountability characteristics

of a cloud service. Any breaches in policies laid down by a service provider should be attributable to an individual user.

Privacy is closely related to confidentiality, it is seen as the active disclosure of sensitive data by the cloud provider to other customers or external sources. Confidentiality is more passive, allowing access via a systemic failure, for example poor security measures.

## 2.4 Service Level Agreements in Cloud Computing

SLA embody the aspects of the Cloud Agreement such as price, security/privacy and legal aspects and provide implementation requirements such as Quality of Service, Geographical distribution, use of Local Clouds to provide Hybrid Cloud solutions [31]. Much work has been carried out on Service Level Agreements (SLA), for example Grozev and Buyya [32] defined a number of characteristics for SLA, which are outlined in Table 2 (below).

Characteristic	Description
Price	Price of cloud resources.
Security/Privacy	Prevention of data access from 3 <sup>rd</sup> parties.
Quality of Service	Definition of service metrics, such as request response times.
Geographical	Where the data is held and processing carried out
Legal aspects	Contractual agreements describing express and implied terms in the contract
Local resources	Use of local resources, such as private clouds or hybrid clouds
Data requirements	Database and data storage issues.

**Table 2 - Service Level Agreement (SLA)**

In addition to the characteristics identified in the CA Quality of Service (QoS) is considered by Abdelmaboud et al. [33], metrics are defined, such as average cloud instance start up time. The researchers found a number of toolsets were being developed to assess QoS in cloud computing platforms.

Geographical aspects of cloud computing impact security and privacy issues [30]. There are also issues with QoS as network latency will be greater if the cloud service is located on another continent [34], however, cloud resources located locally may be more expensive.

The organisation may want to use private cloud resources or combine public cloud resources with private cloud resources to build hybrid clouds. The SLA will need to consider the QoS implications of this approach [35].

Data Requirements for cloud resources, such as database or NoSQL storage could be defined in the SLA.

Mao et al. [36] present a utility based pricing approach that uses a utility linear function. This function maximises utility specified as revenue over time for a set of resources, such as virtual machines, by ranking the execution jobs that must be completed within the specified time constraints. Greenwell et al. [37] introduce pricing based on problem solving semantic description, where a customer is allocated a price based on task requirements, the problem solving method (such as an algorithm) and the knowledge domain (represented by a data store or database).

Carlson [38] describes a cloud security model using a threat based approach, which operates by identifying a number of technical and operational threats, with vulnerabilities in cloud security being exploited to generate threats.

Gonzalez et al [39] present research analysing the security concerns seen in cloud computing. The researchers used secondary sources from academia, organisations such as NIST, The Cloud Security Alliance and the European Union Agency for Network and Information Security (ENISA) and from practitioner organisations. The researcher's findings were presented as a number of high level taxonomy for security issues in architecture, compliance and privacy. The problems with the approach proposed are



firstly, the usage of a taxonomy ignores the rich semantic modelling constructs offered by description logic based semantic description. Secondly, the models described in the research would require additional work to be represented as software based models. Finally, security is strongly connected to other aspects of CSA, such as law, which were not described in the taxonomies.

Geographical location of services is discussed by Buyya et al. [40]. Customers can choose the global cloud service centre they wish to access services from. This will affect the cost and performance of the service. The researchers envisage a service driven by SLAs that are brokered on a geographical basis. Geographical distribution is strongly linked to legal and privacy aspects of SLA, for example the European Union places restrictions on data being held outside its jurisdiction.

Purely legal aspects of service provision are described by Bradshaw [5]. Applicable law is provided by jurisdictions such as US states, the European Union and UK law. Arbitration and variation of contract terms allow disputes to be settled outside a legal system. Suppliers have a number of approaches to contract variation with some having formal processes of variation (in writing) and others having e-mail or web service dashboard notifications.

Grozev and Buyya [32] and Rong [41] discuss the data related aspects of SLA. These aspects are concerned with the amount, transfer time and security of data. The amount of data stored may influence the price a customer pays for cloud services. Transfer time is strongly linked to the co-location of data and data processing units in the same geographic location. Relational database management systems, NoSQL and basic storage mechanisms are offered by cloud service providers [42].

Local Resources (such as organization owned data centres) are described by Toosi et al. [43] are integrated into a hybrid solution with public cloud resources. This requires local resources to form part of the SLA.

## 2.5 Acceptable Use Policies in Cloud Computing

Acceptable Use Policies are identified by CSCC [6]. Table 3 (below) provides examples of terms from a number of AUP [6].

Example Term	Description
Not to install software	The customer should not install software which could harm the cloud service.
Not violate IPR	The provider should not violate the organisations intellectual property
Usage of resources	The customer should not overuse resources.

**Table 3 - Acceptable Use Policy Terms (AUP)**

Bradshaw et al. [5] consider legal aspects of AUP. The researchers cite a number of general conditions that are common to many AUPs, such as prohibition of bulk unsolicited commercial email, fraud, gambling, hacking and hosting of obscene content. Less common prohibitions are use of services for “safety-critical” applications, hosting of materials for specific countries or individuals and the limitation of storage capacity.

CSCC [6] describe a number of prohibitions for AUP. Content based prohibitions include not sending spam and not obscuring or altering e-mail headers. Security related prohibitions are concerned with comprising the security of a cloud service, for example gaining unauthorized access to a service. Integrity prohibitions describe misuse of system resources, such as launching a denial of service attack. Rights of others prohibitions include misappropriating intellectual property.

## 2.6 Requirements Engineering in Cloud Computing

Wind and Schrödl [44] describe a number of approaches to Requirements Engineering (RE) in cloud computing, which were found to be unsuitable in a number of key areas, such as architecture selection, legal issues and

pricing. Cloud services require semantics to express functionality derived from many service providers.

Semantic web-services have successfully used ontologies [45], as have a number of RE approaches [46]. An ontological approach can address some of the shortcomings seen in the current cloud computing RE process, such as lack of completeness, consistency and conflicts between requirements.

Semantic description have been used for modelling requirements for various aspects of information systems. Farfeleder et al [47] describe semantic modelling using natural language for formalising and verifying requirements in embedded systems. Jureta et al.[48] discuss the usage of semantic modelling in stakeholder communication.

A particularly useful semantic modelling RE approach is described by Bogg et al [49]. Bogg explores the use of Problem-Solving Methods (PSMs) expressed as a semantic description in RE. PSM are reusable methods or approaches to problems that can be used across a number of knowledge domains. The approach is seen as cogent for cloud computing, as large compute clouds can be seen in a service brokerage process, which could provide access to a large number of PSMs instances to solve problems across a number of knowledge domains. PSMs can also be instrumented to ascertain resource usage at a notional level to provide expected SLA and guide pricing levels.

Users have problems which can be tackled using a cloud computing, at a given quality of service and cost. Semantic description of requirements is used to support this problem-solving approach. The requirements are modelled as tasks designed to meet specific requirements, problem domains that requirements exist in, and as problem-solving methods which are generic mechanisms to solve problems and bridges between the three elements. The approach enables each user requirement to be considered as a “semantic task”, which can be implemented as a cloud service.

## 2.7 Pricing of Cloud Computing Services

Pricing differentiates cloud computing from previous service technologies, such as non-semantic and semantic web-services. The motivation for concentrating on pricing is the perceived view that cloud computing will move to a utility model for computing [18], where users will access self-service on-demand cloud services.

Kiemes et al [50] discuss a general model for price plans for internet services. Their approach is based on formalisation of work on industrial pricing by Lehmann and Buxmann [51] and Nagle et al. [52]. A pricing model was developed from the literature which was implemented as OWL based semantic description with SWRL rules. The model was developed as standalone description, separate from any business requirements or knowledge domain. Software was developed using the service description to provide rudimentary pricing, for a simple case study, for pricing in a car rental service.

Abhishek et al. [53] discuss the interrelationship between pricing and scheduling on public clouds. They compared on-demand purchase of cloud virtual machine instances with those purchased on an auction (spot market). They found using fixed pricing on-demand cloud instances nearly always provided greater revenue for cloud service providers, rather than a market made up of both on-demand and auction procurement models. The assumptions made in the models such as these need close examination with a requirement for well-formed descriptions of cloud services to establish patterns in the three main service procurement models, reserved instances, on-demand and (spot market) auctions.

Kash and Key [54] describe methods for more sophisticated pricing of cloud computing services. They describe inherent issues in pricing cloud services, such as the requirement to procure across a number of service providers,

the ability to re-price when a service made up of a number of resources changes and, the capacity for prices to change whilst a service is in use. They see an increase in the use game theory to arrive at fair prices.

Di Modica & Tomarchio [55] emphasise the on-demand aspects of cloud computing pricing. They differentiate between direct suppliers of resources and those who provide platform as a service or software (applications) as a service. There is a need for a service level agreement between suppliers and customers and a necessity to provide pricing strategies to obtain an equilibrium between supply and demand. Negotiation protocols allow suppliers to describe services and customers to find the services and, agree a service level and price

Wagner and Sood [56] discuss the economic benefits of building resilient cloud services. The economics of building such systems will feed through to lower pricing of cloud services. The authors propose a system of cyber resilience called Self-Cleaning Intrusion Tolerance (SCIT), this involves periodically rotating virtual machines with clean virtual machine images making servers less vulnerable to attack, with the economic benefits of rotation outweighing the costs. The move to '*Serverless*' cloud computing [57] should see further improvements in resilience and therefore decreases in cost.

## **2.8 Business Aspects of Cloud Computing**

Business aspects of cloud computing have been described by a number of researchers. There is research in to the economics of cloud computing and the general business aspects of cloud computing.

Armburst et al. [9] see cloud computing as a type of utility computing, that is an on demand resource similar to electricity or water. This is seen as changing the economic model of computing. Organisations do not have to invest and manage large amounts capital equipment such as servers. New

organisations can start-up at minimal cost and existing organisations can use the 'elasticity' and self-service aspects of cloud computing to change their computing requirements dependant on the economic conditions.

Weinman [58] describes the characteristics of cloud computing shown in the list below:

- Common Infrastructure
- Location Independence
- Online Accessibility
- Utility Pricing
- On-Demand resources

Common Infrastructure refers to resources being shared from a common pool, similar to the utility concept referred to by Armburst et al as 'utility computing'. Although cloud service providers may have their own management tools or Application Programmer Interfaces (API), users can access infrastructure, platforms and services from a common pool of resources.

Location independence sees the same cloud services offered from a number of geographically dispersed sites, which in turn can be accessed from any geographic location. The main restrictions being network latency, price and legal controls on data storage.

Cloud resources can be easily accessed and managed online. Customer self-service is a key aspect of cloud computing, users can create new virtual infrastructure, platforms or services via a management console or use scripting languages from a command line user interface. The resources can be easily allocated or deallocated.

Utility pricing sees the economic model of cloud computing move from capital based economic model, based on in-house servers to a utility model

based on the ability to purchase services as reserved instances, on demand or through auction markets.

The cloud service user can change their computing needs at will. They may purchase a base level of resources at low cost and then purchase further resources, balancing the cost/benefit of their resource profile to provide the most utility to the organisation.

These aspects of cloud computing provide major benefits to organisations, however, managing cloud resources in a self-service fashion produces a number of difficulties. There are many combinations of resources provided by a number of vendors using a diverse pricing strategies, which must be managed using the best cost to benefit ratio.

Cloud computing is becoming a key component in many organizations' information systems strategy. Ward and Daniel [4] have identified high levels of dissatisfaction with the benefits derived from IT/IS projects, as shown in Table 4 (below).

Benefits Management activity	Level of dissatisfaction
Identification of project costs	43%
Project prioritization	59%
Identify benefits	68%
Development of business cases	69%
Planning the delivery of benefits	75%
Evaluation and review of benefits realized	81%

**Table 4 - Dissatisfaction levels with benefits derived from IS/IT activities**

Organizations can use combinations of hardware and software as required to deliver IS/IT services with some outsourced provision if required.

A number of provision models for cloud computing exist, which are developed from the NIST standards discussed previously [59], Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). The models have common features such as elasticity of usage, flexibility of information storage and user self-service. The differences in the models also influence the view of cloud computing, for example IaaS and some SaaS can be seen as utilities, purchased on price PaaS and some SaaS can be used for business transformation. A number of cloud ownership models have been defined by NIST standards [59] as discussed previously as Public clouds and Private clouds. Hybrid clouds use a combination of public and private infrastructures, bound by some technology that enables data and application portability.

Outside the general business aspects there is a need to discover key drivers in businesses that provide business benefits from cloud computing. The Benefits Management approach has been developed over a number of years by researchers such as Ward and Daniel [4] and Peppard et al.[60]. The approach allows stakeholders to gain maximum business benefit from IS/IT investments by considering the linkage between investments and the business benefits they generate. Ward and Daniel's work [4] shows high levels of dissatisfaction with the benefits derived from IS/IT activities, with 81% of those surveyed having dissatisfaction with the evaluation and review of benefits and 75% having dissatisfaction with the planning and delivery of benefits respectively.

The Benefits Management approach was developed out of a dissatisfaction with IS/IT projects' failure to deliver business value. Benefits Management is defined by Ward and Daniel [4] as "The process of organizing and managing such that potential benefits arising from the use of IS/IT are actually realized". The approach concentrates on benefits delivery, obtaining value from investments and involving stakeholders. There is emphasis on change



management, that is, the importance of IS/IT investments only delivering benefits through organisational change.

Ward and Daniel [4] describe the need for a common language and reference model in exploring benefits enabled by IS/IT investments. Using a semantic description driven approach, multiple stakeholders can develop vocabularies, terms and semantics and map them to form a common discourse. The authors also describe the importance of context, while the semantic modelling and mapping tools help contributors to model context in the Benefits Management process.

## **2.9 Need for Representations**

Now that cloud computing has been defined and the need for more expressive descriptions of the content of cloud service agreements, cloud service requirements, pricing for cloud services and benefits management for cloud service investments has been made it can be seen that there is a need to describe these aspects of cloud services in a uniform and systematic manner. This allows users and vendors to describe their requirements and offerings so that agreements can be fulfilled.

Dillion et al. [61] see the “Cloud Interoperability Issue” as being a major hindrance to the uptake of cloud computing services. Cloud service adopters have to delay their investment in cloud services or lock themselves into a particular vendor's cloud solution. Cloud users need to be able to select and utilise cloud solutions that best meet their needs by having representations that allow information interchange and open selection of cloud services.

Uschold and Gruniger [62] discuss the need for representations as one of the need to communicate concepts between groups with different viewpoints, needs and backgrounds. There may be disconnects or overlaps in concepts which need to be modelled.

Chandrasekara [63] describes the need for common vocabulary and representation, so that stakeholders in any agreement, such as a cloud service agreement can interchange information and negotiate an outcome such as contract or cloud service agreement.

Di Martino et al. [64] see the need for the definition of a common formalism which can completely describe cloud patterns which can be shared among cloud suppliers and cloud users. The common formalism can be used to develop a methodology to recognise similarities in patterns for the purpose of matching user requirements.

A useful differentiation for cloud computing representations, such as service composition, description and pricing are syntactical and semantic approaches.

'Syntactical' approaches use simplistic graphical, mathematical or framework descriptions of cloud services. Examples of syntactical approaches for service composition can be found in Jula et al. [65]. Although the research deals primarily with service composition the issues with syntactical approaches are highlighted. The approaches shown in the list below:

- Graph based algorithms approaches
- Combinatorial Algorithms
- Machine-based methods
- Structures
- Frameworks

Graph based algorithms approaches see the service composition problem as a set of interconnected resources that must be optimised. Optimisation approaches such as linear programming, have scalability issues [66], as the number of cloud resources increase computational complexity increases as to be non-computable in real-time.

Combinatorial algorithms attempt to resolve optimal service composition by examining the different combinations of elements [67]. Combinatorial expansion is a major issue, as the number of possible combinations expand in a multi-cloud environment. To reduce composition time heuristic algorithms may be used to provide possible non-optimal results in a shorter time. Modi et al. [68] used genetic algorithms to prevent intrusion into cloud based systems.

Machine-based methods involve the building Finite State Machines (FSM) [69] and other automata to select appropriate cloud service compositions. These machines are combined with other syntactical techniques such as graph and combinatorial algorithms to provide appropriate compositions. SciCumulus [70] is an example of a FSM for distributing tasks across many cloud vendors. The main issue with the approach is only a few states can be modelled due to increasing complexity and the FSM requires extensive modelling and rebuilding as vendor offerings change. The introduction of spot markets and on-demand cloud instances means this approach is becoming less relevant as frequent changes in market price require the FSM is constantly rebuilt as to make the approach impractical.

Structures are discussed by Sundareswaran et al. [71]. The researchers proposed a cloud service composition/selection approached based on a cloud service provider index (ranking system) based on B-Trees [72]. A B-Tree was built for a number of possible cloud service composition combinations. Service requests were fulfilled by traversing the constructed tree to find optimal service compositions.

Frameworks are prevalent in monitoring of SLA agreements and service composition. Patel et al. [73] developed the Web Service Level Agreement (WSLA) framework for SLA monitoring. The framework comprises three services, a measurement service, a condition evaluation service and a management service. The measurement service samples runtime SLA

parameters of cloud resources. The condition service compares the measurements against SLA values agreed and notifies the management service of any the violations. The management service trigger actions for any service violations, for example, providing additional resources.

Pham et al. [74] propose a service composition framework as an architecture. The service composition architecture comprises a knowledge base which holds information about the current cloud resources in use, a composition agent that uses the knowledge base to service user requests by generating new service specifications and a packaging engine that processes service requests into a delivery of a newly composed cloud services, which are actioned by cloud service providers. The knowledge base will be updated by the composition agent by querying service discovery agents in various cloud services. The packaging engine will use service catalogues in cloud services to obtain detailed configurations of possible cloud services.

It is clear that the (syntactic) approaches discussed in Julia et al. use simple variables and relationships between concepts in the models/approaches proposed. The approaches benefit from using accepted techniques seen in web-service composition and SLA monitoring, however, a common issue are the large combinations and complex relationships between cloud resources, which may be utilised across a large number of cloud vendors.

Semantic approaches deal with deeper meaning seen cloud computing representations. This thesis proposes these semantic based approaches are superior to syntactical based approaches as they provide the ability to compare disparate offerings from a number of vendors and to compare service description elements that are named or structured differently.

There are a number of semantic approaches available, shown in the list below:

- Language based approaches
- Logic based approaches

Staab et al discuss the rise of “emerging semantics” in language processing [75] and other areas such as semantic web-services. It is possible to relate descriptions of cloud services to aid composition of cloud services. Semantic language techniques such as Latent Semantic Analysis (LSA) [76] could be used when a cloud vendor uses their own vocabulary for cloud service agreements. Semantic language techniques would also be useful for developing agreements on a collaborative basis or across different languages.

Although semantic language based approaches including search are useful [77]. Much research has been carried out using semantic based techniques based on logic. Baader et al. [78] describe how logics have developed from simple network models (which have been called ‘syntactical’ in nature in this thesis), similar to UML or entity relationship diagrams. To be developed into logics capable of expressing semantics, but and at the same time being amenable to being processed by computer algorithms (computability).

Levels of formality are key to the success of logics, when semantically modelling cloud computing services and associated agreements. If the logic is not formal enough meaning will be lost. If it is too formal modelling computer based representation may not be possible or take too long. A cloud ‘market’ may compose millions of services from available resources in a short time from a large number combinations. Levels of formality have to be considered when selecting a logic to represent the semantics of cloud services.

## 2.10 Description Logics

### 2.10.1 Introduction

It would be possible to make informal or semi-formal descriptions of cloud computing services [78]. More logically comprehensive descriptions could use formalised models of services using ‘*syntactical*’ modelling tools such as UML [79]. However, these approaches will lose considerable valuable information held in the underlying semantics. However, formalisms seen in UML are highly amenable to development as software systems and, computability is a key requirement when considering modelling techniques for cloud computing services.

Highly formal models can be developed in First Order Predicate Logic (FoPL) [78]. However, the logical description of a simple cloud computing service in FoPL would require several hundreds of clauses and would be time consuming. This is evident in descriptions of simple software programs in formal description language methods, such as Z and VDM [80].

Description Logics (DL) offer a trade-off between quality, expression and computability. Description logics will now be described in detail [78]. DL are defined by Baader et al [81] as a family of representation languages that are used to represent the knowledge of an application domain, in a formal and structured manner. Krötzsch et al. [82] define DL as knowledge representation languages, that form the underpinnings of computer based semantic description modelling languages such as OWL-2 [83]

Concepts in the knowledge domain are described in terms of atomic concepts and atomic roles. Atomic concepts are defined as unary predicates ‘properties’ such as ‘being a student’. An example is shown in Figure 1 (below).

Richard is a student

'Richard' is an atomic class which has the property student, this can be formalised as:

$\text{Richard} \sqcap \exists \text{student}$

Where  $\sqcap$  is an intersection and  $\exists$  is the existential quantifier

### Figure 1 - Atomic Concepts

Atomic roles are defined as binary predicates such as 'writes' (the relation of writing). An example is shown in Figure 2 (below).

Richard is student who writes a document

Which can be formalised as:

$(\text{Richard} \sqcap \exists \text{student}) \sqcap \exists \text{Writes.Document}$

### Figure 2 - Atomic Roles

It can be seen from the two simple examples that semantic value can be extracted from a natural language such as English and can be formalised into logic. This is vital for the modelling of cloud computing service concept modelling, as it allows concepts such as service level agreements and pricing to be formalised and represented in software services. This allows users or autonomous systems to make decisions on cloud service selection.

Description Logics (DL) have a level of expressiveness between Propositional Logic (PL) and First Order Predicate Logic (FoPL). A major advantage of DL over FoPL is decidability, that is, given a set of input parameters a yes/no answer can be provided. It is this decidability that makes DL suitable for building models for semantic intelligence, in areas

such as pricing, composition and benefits managements of cloud computing services.

Reasoning against a constructed DL description can have a clear outcome a useful model can be built in a computerised system that is more powerful than a syntactical model and, is also highly computable.

Computability by reasoning in an acceptable time or in real-time has been the focus of reasoning algorithms, in particular tableau [82] and hyper-tableau [84] reasoning algorithms.

DL have been developed with different levels of expressiveness, defined by the logical constructors they support. Baader et al. [78] describe the DL Attribute Language (AL) as the least expressive DL of practical use. This provides concept description, negation, intersection, restriction and limited existential quantification. The base attribute language can be extended to include a number of additional constructors. Table 5 below shows a set of possible constructors.

Letter	Meaning
U	Concept Union
E	Full existential qualification
N	Cardinality Restrictions
C	Complex concept Negation
H	Role Hierarchy
O	Support for Nominals
I	Inverse properties
R	Reflexivity, Inreflexivity and role dis-jointness
Q	Qualified cardinality

**Table 5 - Description Logic Constructors**



The base attribute language with complex concept negation is named “ALC”. On a practical basis ALC plus support for transitivity (to support inheritance) is seen as a base for many usable DL Languages and is abbreviated to “S”. The Web Ontology Language (OWL) [83] Description Language is described as SHOIN, that is ALC(+transitivity)OIN and OWL-2 [84] is described as SROIQ. Researchers have worked on a number different combinations of constructors.

Grosz et al. [85] describe an earlier language for use of the DAML+OIL as being SHOIQ(D) ALC(+transitivity)OIQ with concrete data types (D). It is possible to develop a description logic with combinations of constructors for a given purpose or for ease of computability.

### **2.10.2 Knowledge Reasoning Built on Description Logics**

Description Languages, based on description languages are used to build Knowledge Representation

TBox [86] (the terminological box) describes the conceptualisation of the universe of discourse in the semantic description. Concepts are modelled with associated properties.

ABox [87] (the assertion box) contains instances of concepts, reasoning takes place on the assertions to see if queries on the semantic description can be satisfied.

RBox [82] (relationship box) defines relationships between concepts.

Tableau [88] algorithms are used in most semantic description reasoning software to ascertain if the ABox assertions can be satisfied, as described above parameters are described in search criteria and a yes/no answer is returned as a set of satisfied assertions. The algorithms build a ‘truth tree’ that is traversed to ascertain satisfiability. In the context of cloud services and their agreements, there could be millions of ABox concepts. This

requires the efficiency of tableau algorithm, which is a major area of research.

A major issue centres on the development of effective and scalable reasoning software resources. Haarslev [89] et al. examine a number of techniques for improving efficiency of tableau algorithms such as deep model (branch) merging and individual model merging. Work on tableau algorithms have made practical DL based ontologies possible.

Motik [90] identifies a number of issues in developing efficient reasoning algorithms. OR branching AND branching introduce complexity into algorithms, with AND branching creating very large models. The researchers have developed a hyper-tableau reasoning algorithm that makes possible to reason against ontologies with large ABox.

Examples of reasoning software are termed 'reasoners' and they are available as standalone software components or as 'plugins' for semantic description editors such as Protégé [91]. Examples of reasoning software can be found in Parsia et al., [92] Pellet reasoner, Glimm et al., HermIT [93] and Tsarkov and Horrocks, FaCT++ [94].

Parsia and Sirin [95] provide a detailed discussion of the Pellet reasoner. They describe how the Pellet reasoner was specifically designed to work with the Web Ontology Language (OWL), an XML based description representation logic based on Resource Description Format (RDF), rather than being a pre-existing Description Logic reasoner which predated OWL. The Pellet reasoner provides a number of features specifically for OWL that make it particularly useful for developing OWL based ontology development.

Firstly, Pellet checks an OWL descriptions meets the restrictions of the OWL language standards which are difficult to ascertain using manual processes. Secondly, it is possible to reason across complex datatypes based on base datatypes such as integers. Support for entailment, a fundamental aspect of

logic that defines the interconnection concerning statements that are consequentially true, is the key interference feature.

Entailment is important for 'Semantic Web' applications, as interconnections and how they are derived is imperative. A number of optimised algorithms have been developed in Pellet, for querying large numbers of assertions in the ABox. Semantic Web applications are characterised by having large numbers of assertions, the algorithms can reduce the search space of queries against developed ontology by discounting non-matches based on different variable types, for example.

In conclusion, Pellet is a reasoner that is specially engineered for OWL and the Semantic Web that supports checking, reasoning and entailment.

### **2.10.3 Applied Usage of Description Logics**

There is extensive usage of description logic ontologies in a number of research areas, notable examples are bioinformatics and software engineering.

A main area of research that utilises description logic based ontologies is bioinformatics [96]. Ontologies such as Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT) [97]. This semantic description provides a multilingual description of clinical terms for reporting in health records. Concepts are organised into categories such as body structure.

Tetlow et al. describes the usage of description logics in [98] software engineering. An example of the usage of description logics is where a software program will have (hidden) semantics encoded in the code that is not explicitly stated by the syntactical language code the program is written in. The semantics of the code can be expressed as description logic based ontology.

#### 2.10.4 Description Logics in Requirements Engineering

Ontologies provide a structured framework for modelling the concepts and relationships of a domain of expertise. Ontologies support the creation of repositories of domain-specific reference knowledge [99]. Ontologies have been used for requirements engineering for a number of years. Zave and Jackson [100] described “core” semantic description as solving the “Requirements Problem”. The core semantic description established the minimum set of information required for engineering requirements as:

$S, W \vdash R$

Given:

**R** are given requirements

**S** is a complete specification

**W** are domain assumptions

Proof of Obligation requires that the specification and domain assumptions to be satisfied by the requirements [101]. This points to a “pure” but simplistic approach to RE that only specification and domain assumptions are required in the RE process. The approach is criticised, by Jureta et al [48], who state that partial requirements cannot be described in Zave and Jackson’s model, and only a complete specifications can be created. The requirements specifications cannot be ranked in terms of better or worse requirements for a given specification. Non-core requirements cannot be defined and, nice to have requirements may be lost.

Castanada et al [102] identify a number of benefits in using semantic description in the RE process. A requirements model is imposed enabling the structuring of requirements and the knowledge domain in question. The Interrelationships between requirements can be defined.

A number of attempts have been made to specify a semantic description to describe the components of cloud computing, a typical example being Youseff et al [103]. These ontological approaches suffer from viewing cloud computing as a continuation of Software as a Service and concentrate on low level virtualisation.

Each user requirement can be defined as a semantic task, this facilitates enhanced capability in the validation of specification, the discovery of services and composition of cloud services. Cloud computing can be seen as more complex than traditional Information (IT) environments. User requirements are expressed at a high level, a brokerage layer or service will find and price these requirements from a number of cloud computing resources. Cloud computing resources will then execute tasks for these brokered requirements.

## **2.11 Description Logics in Cloud Computing Service Specification**

There has been some research into using description logics in using description logics in cloud computing service specification. These have been limited to the items shown in the list below:

- Service Composition
- Service Level Agreements (SLA)
- Requirements Engineering
- Security

### **2.11.1 Service Composition**

Amato [104] [105] Used a pattern based approaches based on formal models to compose cloud services from a number of resources provided by multiple vendors.

Fang et al. [106] [107] developed a semantic framework, using description logic based semantic description. The researchers used OWL2 and with Fuzzy extensions to describe cloud service components. Using fuzzy search researchers were able to develop descriptions, to retrieve service components and produce recommended service compositions. The components for compositions could be selected across a number of service models (IaaS, PaaS and SaaS).

The key driver for developing the approach was to achieve a high degree of agility, that is, to respond to changes in user requirements and feedback from cloud service providers. This allows service composition to be adjusted quickly.

A prototype semantic description and software toolset was developed, by performing experiments and, it was found that an 'agility score' for a given service composition could be synthesised and, the system was able to recommend compositions, providing recommendations in plain English.

Ghazouani and Slimani [108] undertook a survey of cloud service description. The researchers take a broad view of cloud service description, considering the technical, operational and business aspects of cloud computing. They found widespread use of semantic techniques in cloud service description and using Unified Service Description Language (USDL) as a suitable semantic description language to describe all aspects of cloud computing. Oberle et al. [109] describe USDL as semantic service description language based on description logics, designed to describe both human and machine based services.

### **2.11.2 Service Level Agreement Specification**

Dastjerdi et al. [110] [111] propose a semantic service level agreement description approach that can be used by multiple cloud stakeholders (providers, users, managers and academics). The researchers built a SLA

model using the Web Service Modelling Ontology (WSMO) to prevent SLA failures promulgating through a cloud service when SLA policies are violated. An architecture was created for cloud services, the components of which are shown in the list (below).

- Discovery – Finding service components
- Ranking – of appropriate cloud services
- Coordination of cloud services
- Monitoring of cloud services

The researchers also investigated algorithms to find suitable candidates suitable for discovery, ranking and monitoring of services.

It was found that SLA to cloud resources could not be achieved by simple pattern or 'syntactical' matching and, that a semantic approach based on description logic was required to achieve matching at a more meaningful level.

Joshi et al. [112] describe automating cloud service SLA using semantic approaches. Semantic models were developed for cloud service requirements, discovery, negotiation, composition and consumption. The main requirement for using a semantic approach was to allow distributed and disparate cloud resources to automate the acquisition and consumption of the resources. If a semantic approach was not used it would be impossible to describe resources and SLA across multiple vendors.

A Description Logic (DL) based on semantic description and toolset were developed to allow naïve users to work with cloud SLA from requirements through to consumption.

### **2.11.3 Requirements Engineering**

Immonen et al.[113] developed a digital services environment focused on description logic based semantic description. The members of environment comprised cloud service providers, brokers and consumers. The members worked of a description logic based elements, which aided Requirements Engineering (RE). The main findings from the research are shown in the list below:

- A domain model provides concepts of the domain and relationships as description logic semantic description
- Knowledge management model describes knowledge and design patterns used in the business as description logic semantic description
- Service engineering documents the RE techniques

Takabi et al. [114] see description logic based semantic description as a solution for the defining requirements across a number of different cloud resources offered by an increasing array of providers. The researchers see the most applicable areas being requirements for privacy and security.

### **2.11.4 Security**

Souag et al. [115] describe a description logic based semantic framework for cloud security requirements engineering. Security is seen as a major concern in cloud computing, especially in the area of public cloud usage.

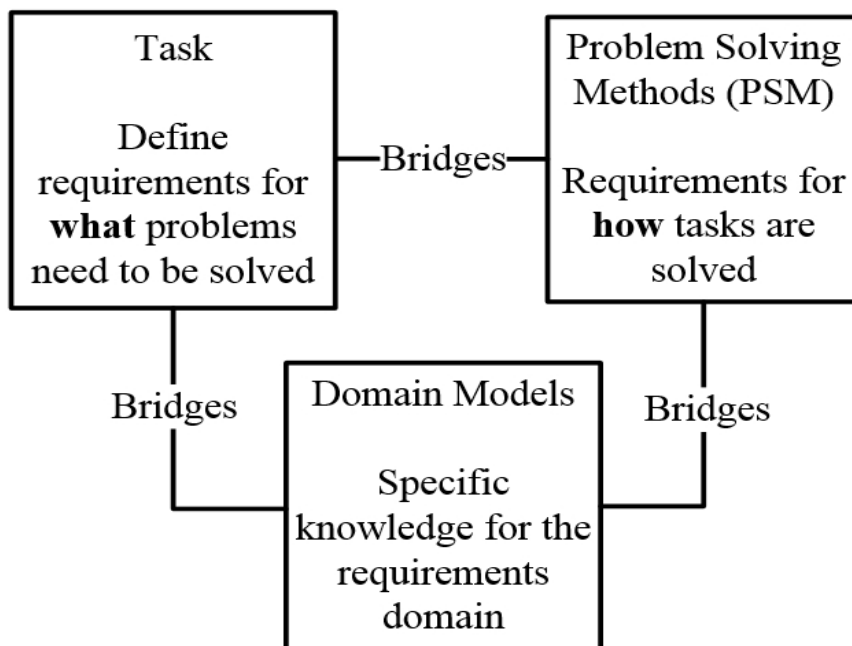
The researchers see knowledge reuse as the major reason for using DL based semantic description. The specification of semantics in threat assessment is seen as the main application area for DL based semantic description.



Tsai et al. [116] discuss DL based semantic description for role based security in cloud services, especially when provisioned across a number of cloud vendors. The semantic description based in DL allows the simplification of role specifications.

## 2.12 Problem Solving Semantic Description

A key aspect of the research undertaken is to examine a particular class of semantic description for use with cloud computing specifications. This class of semantic description is described as Problem Solving Ontology (PSO). This semantic description allows other ontologies such as pricing) to be “overlaid” and provides a framework for these ontologies. Figure 3 (below) describes the components of Problem Solving Ontology



**Figure 3 - Problem Solving Ontology**

Fensel et al [45] describe the Unified Problem-solving Method Development Language (UPML) is a framework for developing knowledge-intensive reasoning systems based on libraries of generic problem-solving

components. They go onto describe the UPML architecture, shown in the list below:

- A *task* that defines the requirements for problem that is to be solved
- A *problem-solving method (PSM)* that defines the reasoning process
- A *domain model* that describes the domain knowledge of the knowledge-based system
- Bridges are used to map and define the relationship and transformation between the task and PSM.

Figure 3 (above) shows the interrelationship between the UPML architecture components. Bridges are used to map each of the components, for example a task “compute overall cost” (what is required) may require a PSM defined as an algorithm to fulfil a task (how a task is achieved ), such as “compute best price”. The bridge would define the relationship and transformation between task and PSM.

Crubézy and Musen [99] describe how Problems Solving Methods (PSM) and (Domain) Ontologies can be combined to produce knowledge systems. Musen [117] describes Domain ontologies as “Characterisation of concepts and relationships in an application area, providing a domain of discourse”. Domain ontologies define the knowledge specific to a problem, for example information on characteristics regarding pharmaceuticals and their side-effects, if requirements for drugs prescription were being defined. This is a logical separation of from PSM which are generic methods that can be used to solve a number of problems in different domains. Tasks bring together Domain Models and PSM. This separation provides greater reuse of requirements.

Given that PSM themselves can be described by semantic description, a purely semantic description based approach to knowledge systems and requirements engineering can be produced. PSM, domain models and

bridges that map requirements between each component. This allows requirements to be defined using semantic description modelling tools.

Expressing requirements using a problem-solving semantic description allows the requirements engineer to utilise an approach that is well suited to the cloud computing environment. Tasks can be seen as a unit of work that is well understood by users. Problem Solving Methods (PSMs) can be seen as reusable specifications for solving the problems posed by tasks. Domain models can be built as a semantic description, so it can be understood by users and verified using ontological reasoning tools. The requirements semantic description can be seen as a specialisation of more generalised problem-solving semantic description, such as the Unified Problem-solving Method Development Language (UPML).

The main issues with developing a semantic framework are shown in the list below:

- Potential researchers or users have to learn description logics and problem solving ontology
- Effort and time to generate concepts and relationships
- Effort and time to gather and insert assertions based on concepts and relationships
- Editing and checking semantics for correctness

### **2.12.1 Origins of Problem Solving Ontology**

Mizoguchi, et al. [118] discuss task orientated ontologies as “Ontologies that can separate tasks from the knowledge domain, so the tasks can operate independently, leading to task reuse”. The approach of building problem solving ontology is strongly related to linguistic analysis of task specifications, identifying tasks and problem solving methods in verbs and domain knowledge from linguistic constructs.

Gomez-Perez and Benjamins [119] describe Problem Solving Methods (PSM) as a “valuable components for constructing knowledge based systems (KSBs)”. PSM provide constructs that are defined as description logic ontology which guide ontology developers in analysing and formalising real-world problems. Reuse of knowledge, for example reuse of PSM is seen as a key benefit.

Yan et al. [120] see the roots of PSO in approaches such as the TRIZ methodology [121], a Theory of Inventive Problem Solving. This is seen a heuristic approach to decomposing problems, to identify tasks and understand how tasks can be reused independently (as problem solving methods) from knowledge domains. The researchers developed a description logic based ontology to support the TRIZ method, which was a effectively a meta-semantic description for problem solving (a problem solving semantic description for problem solving semantic description).

### **2.12.2 Unified Problem-solving Method Development Language**

Fensel et al. [45] developed the Unified Problem-solving Method Development Language (UPML) which is based on description logic based semantic description and was developed to provide reusability for architectures of reasoning aspects of knowledge based systems. The aim is to provide catalogues of reusable knowledge components.

The UPML approach provides a highly graphical approach to specifying problem solving semantic description and implements the pattern of task, problem solving methods and domain models with bridges between the three concepts.

Problem solving methods are seen as reusable concepts that can be applied to a number of domain models. In cloud computing problem solving methods can be seen as cloud resources, domain models are application/business

areas, tasks employ problem solving methods that act against domain models to solve problems.

Scharffle et al. [122] emphasize the importance of UPML when aligning ontologies from multiple sources. The patterns and analysis based approach embodied in UPML allows ontologies to be matched at a semantic level, leading to the ability to identify useful problem solving methods from a number of disparate knowledge domains. Alignment of cloud service agreements and contracts between cloud service vendors and customers are areas where UPML are applicable.

## **2.13 Discussion and Conclusions from Literature Review**

Cloud computing has become an established and mature technology in many organisations with well-established definitions of delivery models (such as infrastructure, platforms and services) and ownership models (public, private and hybrid). The literature review introduced the four areas of research of interest, which are interlinked and provide the basis for the approach to the semantic intelligence cloud, are shown in the list below:

- Cloud Service Agreements
- Requirements for Cloud Services
- Pricing of Cloud Services
- Benefits Management of Cloud Services

Although, these areas of research are not exhaustive, they provide fundamental concepts required to manage cloud services and represent areas of research not examined by other researchers in detail using semantic techniques, are shown in the list below:

Cloud Service Agreements (CSA) provide three fundamental artefacts identified from the literature, shown in the list below:

- Customer Agreements (CA) – What the customer requires from the cloud service and what the cloud supplier is willing to supply.
- Service Level Agreements (SLA) – Metrics for each cloud service requirement that the cloud supplier must meet
- Acceptable Use Policy (AUP) – Penalties if the customer breached part of the agreement with the cloud service supplier

It was found from the literature that many researchers concentrated on SLA to the detriment to the two other artefacts. The modelling techniques used to model SLA were very simplistic in nature, i.e. they used unsophisticated mathematical or “syntactical” models. Research has also concentrated on simulations of low level SLA, for example availability and response times on virtual machines on infrastructure as a service.

Cloud Service Agreements provide the starting point for examining cloud services and it was found there are strong linkages to other aspects of cloud computing, such as contract and legal aspects of cloud computing, cloud service requirements and cloud service pricing.

The literature review continued with a consideration of requirements for cloud computing services. It was found that a number of researchers have considered requirements for cloud computing services, but there was little research into using semantic techniques for defining and modelling cloud service requirements.

Research into semantic web-service requirements provided a good area for literature review, as many of the issues seen in semantic web services are similar cloud computing services, for example, composition, orchestration and deployment have similar requirements in web-services and cloud services. However, cloud computing services have some unique challenges. Self-service aspects of cloud services require a user translates their requirements and CSA into physical cloud services, by examining many possible service combinations and configurations. Pricing is a major area of

requirements in public cloud computing services, whereas semantic web-services have usually been hosted on private infrastructure or platforms or are “free” public services.

Cloud service providers have introduced complex pricing structures, based on time, resource usage and geographic location. Traditionally suppliers of public cloud services have had fixed price reserved resources and on-demand pricing. This has been augmented with spot or auction pricing models. A user must decide the most cost effective combination of pricing models, this will change in real-time and from feedback deployed resources. To obtain optimal pricing a sophisticated user will require a decision support system or service. Even with a decision support methodology, a simplistic service modelling technique may be unable to provide optimal pricing in real-time due to combinatorial expansion.

In a similar manner to research into CSA, requirements researchers have concentrated on simplistic mathematical or “syntactical” models for pricing, much research has been carried out for micro-analysis of pricing in markets using mathematical models. These models are of use to cloud service consumers in the long term, however, offer less value in the short term decision making. Syntactical approaches also have issues when modelling the large number of service combinations and configurations and modelling semantically equivalent concepts. Concepts used in individual providers’ pricing structures are syntactically different, but semantically similar.

Business value provided by cloud service consumption research has concentrated on economics and business analysis. The literature review focussed on the Benefits Management approach, which is a robust academic framework which has been applied to information systems.

The literature review then continued with an evaluation of the deficiencies in current approaches to cloud service specifications and, the need for better representations. The case for the usage of semantic representations over

“syntactical” was presented, the major benefits being, are shown in the list below:

- Simplistic “syntactical” or mathematical representations do not capture all requirements
- Large numbers of configurations and combinations seen in cloud service representations cannot be represented without rich knowledge representations and reasoning
- Concepts seen in cloud services are semantically similar, but syntactically different. It is easier to compare services at a semantic level as common concepts exist in cloud services provided by a number of suppliers
- Semantics can be used to link different users’ views of requirements and service provision

A number of syntactic approaches were described and found to be deficient, suffering from the factors described above. Semantic approaches were introduced as language based approaches and logic based approaches. Language based approaches have some use, for example semantic matching of documents and concepts, however, the usage of logic based approaches is seen as the key technique when semantically modelling cloud services.

Levels of formalism are seen on a continuum from simplistic natural language descriptions, which can be imprecise and ambiguous, to full formal specification in First Order Predicate Calculus. The issue with full formal specification being the amount of specification required and level of formalism can be onerous even for the simplest cloud service.

Description logics offer a middle ground between imprecise specification and full formal specification. This allows an appropriate level of specification with an acceptable level of formalism, that most importantly is amenable



reasoning and computability in real-time. The logical constructors in description logics can be tailored to requirements of users or knowledge domains. There are well-supported set of standards and toolsets for Description Logics.

Problem Solving Ontology is a specialisation of Description Logic based semantic description that represents activities as tasks, this is seen as appropriate to cloud computing service specification, as each user requirement can be modelled as a task, which is a convenient concept all cloud service users should be able to comprehend.

Tasks can be recursively be decomposed into further (sub) tasks. Tasks are fulfilled using generic Problem Solving Methods (PSM), which can be analysed (instrumented) to define their notional resource usage such as CPU and memory usage which can be mapped to cloud service choices and pricing. The Unified Problem-solving Method Development Language is a problem solving meta-semantic description that can be used to model cloud services.

In conclusion current approaches to cloud service specification have been found deficient in two ways, are shown in the list below:

- In content – there has been a narrow focus on physical specification of infrastructure and service level of agreements. There are further aspects of agreements that necessitate further examination. Requirements, pricing and benefits derived from cloud services also require investigation.
- In representation – Syntactical and mathematical representations are not powerful enough to model cloud services, given their complexity, possible combinations and configurations and different syntactical representations used by cloud service providers. Full formal specification of cloud services

would be difficult due to the amount of effort required for full formal specification and would be onerous to compute in real time.

Description logics and the usage of Problem Solving Ontology are seen as particularly suitable for modelling cloud services. The task orientated approach views the task as a convenient unit of work, which service users and providers can relate to.

## **Chapter 3      Related Work**

### **3.1 Chapter Overview**

Related work expands the literature review to technologies that could be used to provide semantic representations for the proposed framework. Technologies developed for the “semantic web” which has developed over a number of generations is seen as an important starting point.

A number of semantic technologies are examined in detail along with examples of their usage.

### **3.2 Introduction**

Much of the research into semantic frameworks in software systems has concentrated on trying to incorporate semantics into the World Wide Web (WWW). This related work provides a context for concepts and technologies useful in cloud computing. The discussion starts with Web 2.0 and Web 3.0 which are terms which have been defined by researchers for making WWW less “syntactical” and more semantic. There is a look forward to technologies that build on Web 3.0 and is named “Web 4.0”.

How Web 3.0 will be achieved is then described by considering how semantics are incorporated into web technologies. Some examples of semantics used in web-technologies are then provided.

In the following section technologies related to cloud computing are discussed.

The final section describes how semantic frameworks are realised as ontologies and as toolsets to support semantic description development.

### **3.3 Web 2.0**

Boulos and Wheeler [123] discuss the enabling social aspect of Web 2.0 technology over the first generation of WWW software systems. An example of how the knowledge domain of health care has been enhanced by allowing clinicians, patients and others to interact via collaborative services, social search engines and other technologies. The “Wisdom of Crowd” technologies such as Wikipedia, is highlighted as an atypical Web 2.0 application.

This architecture of participation can be implemented on traditional web technologies, as much of the cognitive processing of information is carried out by the social participants. The major criticisms of Web 2.0 technologies is that the knowledge presented has no peer review process, as seen in academic journals, there is much ‘noise’ in terms of pointless or spurious information, which must be filtered by users, at a cost of their time and, there is no deep understanding of knowledge presented.

Brown and Adler [124] describe the social learning aspects of Web 2.0 technology “Different groups of learners can be brought together to unleash productive inquiry. Niche learning opportunities can be provided as costs of learning in different combinations can be facilitated”. A number of organisations can be brought together to provide a learning experience. However, the learning experience can only be seen as deep by the participants in the social network, the “Web Technology” is still a delivery or presentation mechanism in the same way a traditional web platform is. An increase in the cognitive ability of the delivery platform could provide a better learning experience.

Churchill [125] discusses the usage of Web 2.0 technology in education. The usage of blogs in education by teachers and students are seen as useful tools in the education experience, as passive readers and active

contributors. An interesting finding of this research was maximisation of the value of blogs required augmentation by other social media techniques such as syndication and tagging. This seems to point towards the emerging concept of Web 3.0, where number systems thinking approaches and reductionist approaches are combined, to create added value or make the information manageable by usage of statistical techniques such as trending or clustering.

Constantinides and Fountain [126] describe Web 2.0 as a collection of open-source, interactive user-controlled online applications that allow users to participate in sharing of experiences and knowledge. They identify five categories of application blogs, social networks, communities, forums and content aggregators. A major finding is that Web 2.0 is a concept rather than an integrated set of tools, information can only be shared by applications at a very simplistic level, for example integrating a number of news feeds using syntactical filters.

Silva et al [127] provide an historical narrative for the transition from Web 1.0 to Web 3.0. Web 1.0 is seen as set of mainly static web pages with little user interaction, users were seen as passive information users. Web 2.0 provided the ability of users to create their own content such as blogs and to interact and possible change the content provided by other users. The researchers describe Web 2.0 as tool focussed, tools such as syndication tools, provided the driving force for changes in users behaviour. Web 3.0 is described as providing ubiquitous and pervasive content and services

### **3.4 Web 3.0**

Silva et al. describe the features of Web 3.0. Ambient Intelligence is defined as “the convergence of ubiquitous computing, ubiquitous communication and interfaces adapting to the user”. Smart Interfaces can be seen as user interfaces that adapt content and presentation to target a specific user.

Intelligent agents are used to infer the semantic meaning from the content of existing web pages.

The implementation of the semantic web faces two major challenges. Firstly, to link existing content to semantic meaning by using metadata, this approach has been used in Semantic Web Services, which are used in ontologies for problem solving (discussed in Crubézy and Musen [99] in the literature review). The second challenge is to create applications that use the metadata; much research development work has been carried out into development of semantic web services. An even bigger challenge would be to use machine learning create metadata and applications that utilised the metadata.

Lassila and Hendler [128] recognise the difficulty in identifying an outright definition of Web 3.0 and, propose that it is synonymous with the semantic web. They identify key technologies, such as the Resource Description Framework (RDF) and Ontology Web Language (OWL) becoming de facto standards, which allow semantics to be represented embedded in existing web technologies.

Organisations out with the research community have started to use standards such as RDF, along with the query language SPARQL. The move of these technologies into mainstream computing will increase the usage of semantic web-technologies.

Hendler [129] describes Web 3.0 being built on Web 2.0 technology with semantic description languages such as RDFS and OWL providing semantic mark-up. This is a very data centric approach, with the main emphasis being on merging information from multiple data sources. This is a very practical approach as it builds on existing technologies. The approach could be implemented on existing cloud platforms and use relational database management systems. However, it requires a software developer to set up

linkages manually and, therefore relies on the cognitive power of humans and, thus cannot really be considered a true Web 3.0 approach.

García-Crespo et al. [130] discuss the usage of Web 3.0 concepts in digital libraries using their CallimachusDL digital library. This library integrates social web and multimedia elements in a semantically annotated repository. They describe the semantic web as automated information access based on machine processable semantics of data. This approach ties into the concept of the semantic intelligence cloud. The concept being information access based on machine-processing i.e. higher machine cognitive processing and/or human intelligence input.

Researchers have started to use social web information from Web 2.0 in a reductionist fashion as suggested by Web 3.0. Russell (2011) describes “Mining the social Web”. A number of social media sources were analysed using statistical techniques to derive new knowledge.

### **3.5 Web 4.0**

Nath and Iswary [127] discuss Web 4.0 and see an increase in the usage of semantics combined with machine intelligence providing increasingly personalised information for users. The aspects of Web 4.0 are, are shown in the list below:

- Increased usage and provision of Natural Language and Understanding through semantics
- Greater usage of Machine to Machine (M2M) communication
- Mobile interface usage

Now that an outline for what is required for an improved “web”. The discussion will move on to how it can be achieved.

### 3.6 Modelling the Semantic Web

The semantic web was defined by researchers Allemang and Hendler [83], with reference and contrasting with the World Wide Web (WWW), which is seen as (purposely) chaotic system, are shown in the list below:

- Anyone can say Anything about Any topic (AAA)
- An open world assumption – more information is being added
- No unique naming – Similar concepts are described using different structures and languages
- Network effect – Growth driven by networks of people, creating more and more growth
- A data wilderness – loss of information as it is unreachable, cannot be found or understood

The HTML language is designed for presentation language rather than a language for storing and exposing knowledge [132]. The linkages between webpages are 'syntactical', it requires search engines crawling the web or data mining to find linkages based on simple syntactical representation or mathematical formula. Many interesting relationships between concepts on the WWW are lost or obscured and information seen by users is that presented by tools such as search engines and not all possible information. Researchers into the semantic web see modelling as process for solving the problems of lost information, lost linkages and lost semantics.

Stakeholders can collaborate using models, models can incorporate different stakeholder worldviews and models can be audited to explain why a conclusion was reached. Considering semantics, going beyond simple syntactical models and into the deeper meaning of models allows more of the 'knowledge' to be captured.



Hartig [133] describes the need to link data on the WWW with well-defined semantics, which allows users to find related information more easily using semantic query languages.

Early approaches to adding semantics to WWW has seen embedding of semantic mark-up languages to existing web-pages, to establish semantic frameworks and ontologies across the web. Patel-Schneider and Horrocks [132] describe two modelling paradigms which are being used in the semantic web, the classical paradigm and the datalog paradigm.

The classical model uses formal semantic description and modelling languages such as RDF and OWL and present a formal description logic of semantic representations of knowledge. A formal TBox, RBox and created for the semantic description with rules defined in languages such as SWRL and queries are formalised into languages such as SPARQL and fuzzy variants of query languages. Facts are not limited by any rules, this is called an 'open world assumption'. In an open world assumption where what is not true is unknown. This contrasted with the closed world assumption, where the unknown is either true or false.

The Datalog Model is based on a declarative/deductive database programming approach, the semantics of models are limited to rules provided by the datalog environment. Facts are limited by rules, in the same way a relational database can only produce query results from what is present in a database, in terms of data and the relational algebra applied to it. Both the datalog model and relational databases operate on a closed world assumption.

The classical model is more suited to open, unstructured environments without unique naming, where there may be multiple interpretations (often described as *Weltanschauung* the German for 'world view') of the same concept. An important feature of the classical model is to reason against semantic description to produce new knowledge. The negative issues with

the usage of the classical model, are gaining knowledge of logics, such as description logics, combined with domain knowledge. An ontology developer will have to become skilled with semantic description editing tools, which may be unfamiliar. Processing times for semantic description queries may be an issue when developing large ontologies.

The Datalog approach is suited to well defined and constrained knowledge domains, where semantics can be tailored to the knowledge domain. The concepts and technologies used in Datalog are familiar to database developers and users. Processing times of datalog queries will be acceptable due to lack of complexity and the constrained semantics of the approach. This is evidenced by the usage of datalog analytics seen in 'Big Data' applications [11] [134].

### **3.6.1 Semantic Mark up**

Semantic mark-up is a methodology for annotating or augmenting WWW webpages and web-services. A number of mark-up approaches have been observed, such as Microdata, Microformats and Resource Description Framework (RDF) based standards.

Microdata [135] is used to build "semantic" information into existing HTML based web pages. This information can be parsed by WWW infrastructure, such as search engines to provide better search results. The approach benefits from building on existing HTML mark-up and thus is familiar to those developing software using this technology. The main issues with this technology lie in the fact HTML is a presentation technology and, the semantics rely on the usage of specialist vocabularies and schemas and specialist processing technologies to extract "semantics" from the HTML. This requires the definition of the semantics are split between the mark-up, information defined by the developer and, schemas and their interpretation. An example of Microdata mark-up is shown in Figure 4 below, taken from Hickson [135].

```

<section itemscope itemtype="http://myvocab.example.org/animals/cat">
  <h1 itemprop="name http://example.com/fn">Hedral</h1>
  <p itemprop="desc">Hedral is a male american domestic
  shorthair, with a fluffy <span
  itemprop="http://example.com/color">black</span> fur with <span
  itemprop="http://example.com/color">white</span> paws and belly.</p>
  
</section>

```

**Figure 4 - Microdata Example**

Each property is marked by the “itemprop” tag.

Microformats [136] are an HTML semantic mark-up similar to Microdata. They also embed semantic information in HTML and therefore have the same advantages and disadvantages as Microdata. An example of the Microformat mark-up taken from Luo et al. [136] is shown in Figure 5 below.

```

<div class="service" id="svc">
  <h1><span class="label">ACME Hotels</span> service API</h1>
  <div class="operation" id="op1">
    <h2>Operation <code class="label">getHotelDetails</code></h2>
    <p> Invoked using the <span class="method">GET</span>
    at <code class="address">http://example.com/h/{id}</code><br/>
    <span class="input">
      <strong>Parameters:</strong>
      <code>id</code> – the identifier of the particular hotel
    </span><br/>
    <span class="output">
      <strong>Output value:</strong> hotel details in an
      <code>ex:hotelInformation</code> document
    </span>
  </p>
</div></div>

```

**Figure 5 - Microformat Example**

XML based formats based on the “Classical Approach” and “Datalog” described above represent “true” semantic mark-up. XML based mark-ups are separated from the HTML presentation mark-up. The advantages of XML based mark-ups is they provide the ability to specify the full semantics in formats that are familiar to semantic framework and ontology developers. The disadvantages of the XML based formats are web-developers have to learn new languages and concepts, when they may only want to describe the properties and simple linkage of some concepts.

The Resource Description Framework (RDF) provides the base for many XML based ontology mark-ups, a principal ontology description format Ontology Web Language (OWL) is built on RDF. RuleML is the major XML mark-up built on the principles of Datalog. However, with the popularity of JavaScript based technologies, software developers requiring a simple mark-up for linked data has seeing interest in Java Script Object Notation Linked Data (JSON-LD).

This section will continue with a more detailed examination of RDF. OWL and JSON-LD.

### **3.6.2 Resource Description Framework (RDF)**

The Resource Description Framework (RDF) [83] is a general purpose XML based standard, used go represent semantic metadata. The standard uses a ‘triple’ statement to represent a relationship, shown in Figure 6 below.



(S(ubject), P(redicate), O(bject))

**Figure 6 - RDF Triple**

An RDF schema can be used to represent classes and hierarchies. A main strength of RDF is the ability to merge data from two or more data sources. RDF sacrifices small document size for this ability to merge easily. Unique identities are implemented as Universal Resource Identifiers (URI) so that any resource being modelled can be accessed over a semantic description or semantic descriptions that model resources over the WWW or other systems.

The RDF standard only specifies the XML specification and not how it can be processed. It is possible to build specialist representations on top of RDF and to build commercial strength databases, such as graph based databases, which use technologies such as RDF, to model relationships between subjects and objects as predicates. These representations and products will be discussed in detail later.

Breitman et al. [137] emphasise RDFs power in representing metadata and see it as a base language to support semantic description development and to support information exchange that can easily be read by machine based systems. A semantic description fragment described by Breitman et al. is shown in Figure 7 below:

```
<cs:Book
  rdf:about="#R20301">
  <dc:creator
    rdf:resource="http://www.cat.com/auth#R051156"/>
  <dc:title>
    SEMANTIC WEB: CONCEPTS AND TECHNOLOGIES
    FOR THE GEOGRAPHIC INFORMATION SCIENCES
  </dc:title>
  <dc:date rdf:datatype="&xsd:date">
    2021-01-20
  </dc:date>
  <cs:noPages>
    324
  </cs:noPages>
</cs:Book>
```

**Figure 7 - RDF Ontology Fragment**

The RDF standard [138] identifies the three object types in RDF, shown in the list below:

- Resources – Identified by URIs, a “book” in the example above
- Properties – An attribute or characteristic to describe a resource, a “book title” in the example above
- Statements – Combinations of resources, properties with actual values, formally defined as subject, predicate and object as stated previously

Pan and Horrocks [139] show how a RDF schema can be built on to resolve semantic problems inherent in the RDF schema metamodeling. A number of modelling languages are built on RDF, one of most important languages is Ontology Web Language (OWL) which will now be discussed.

### **3.6.3 Ontology Web Language (OWL)**

OWL and OWL-2 [92] [84] allow the modelling of description logic based semantic description and are built on the base RDF standard.

Allemang and Hendler [83] describe the key functionality of OWL as being able to create restriction classes. These classes allow the exclusion of some member classes that don't apply to all members of a set of classes, restrictions can also be driven by class properties. Restrictions can be used to build complex relationships by reasoning through inheritance.

OWL Implements a full range of logical operations required to implement Description Logic based semantic description described in Chapter 2, using TBox, RBox and Abox, are shown in the list below:

- Unions
- Intersections
- Cardinality (1...N) relationships

Using OWL it is possible to implement Description Logic based semantic description on the WWW, in web-services or as standalone semantic description. The XML RDF based format is quite verbose compared to a binary format, however, XML can easily be transmitted and processed and the merging of OWL based ontology is easily achieved.

#### **3.6.4 JSON-LD**

Hitz [140] describes JSON-LD which is based on the JavaScript Object Notation (JSON) and thus the extending the language for Linked Data (LD) hence the LD extension. The JSON notation is used to model data on many websites and web-services. JSON-LD is built on JSON to allow the definition of linked data, across websites, it can be seen as a graph definition language. The linked data can be queried in a similar manner to OWL, with queries expressed as JSON.

The JSON-LD to RDF API allows specifications written in the two formats to be interchanged, allowing a choice of mark-up and, both formats can be used to express similar semantics. The choice between XML (OWL) and JSON standards will depend on the 'heritage' of a semantic description developer, the XML standards have a longer history of being developed from academic research. The JSON standards have been developed from software engineering practice.

Chalk [141] showed how JSON-LD could be used to build semantic description for scientific data. The Figure 8 below taken from Chalk [141] shows an ontology fragment, translated from an OWL.

```

{
  "@context": [
    {
      "sci": "http://stuchalk.github.io/scidata/ontology/scidata.owl#",
      "@vocab": "http://www.w3.org/2001/XMLSchema#",
      "scope": {
        "@id": "sci:dataScope",
        "@type": "@id"
      },
      "quantity": {
        "@id": "sci:quantity",
        "@type": "string"
      },
      "property": {
        "@id": "sci:property",
        "@type": "string"
      },
      "value": "sci:numericValue",
      "number": {
        "@id": "sci:number",
        "@type": "float"
      },
      "unitstr": {
        "@id": "sci:unitOfMeasure",
        "@type": "string"
      }
    }
  ],
  "scope": "molecule/1",
  "quantity": "magnetic field strength",
  "property": "Anisotropic NMR Shielding",
  "value": {
    "number": 3.2,
    "unitstr": "ppm_Hz"
  }
}

```

**Figure 8 - Example of JSON-LD**

### 3.6.5 Triples Databases

Triples databases are used to store RDF style relationships in a database which can be queried by languages such as SPARQL. This allows ontologies to be represented in a manner suitable for commercial development. Triples databases are highly optimised processing queries and support transactions and serialisation to maintain ACID (Atomic, Consistent, Isolated and Durable) criteria required for commercial data processing.

Urbani et al. [142] demonstrate the ability of triples databases to process RDF and OWL triples in the WebPIE architecture, which was able to reason over one billion triples in a few hours. This level of processing time would be required to process cloud service combinations to achieve optimum service level agreements and pricing in real-time.



## **3.7 Semantic Rule and Query Languages**

Semantic rule languages allow the specification of rules in simpler manner than description by the logical constructs in the semantic specification being used. Semantic query languages allow queries to be run against a semantic specification.

### **3.7.1 RuleML**

RuleML (Rule Mark-up-Language) was developed as a collaborative semantic description development to provide an open standard for an RDF derived rule language [143]. The key driver for development of the language was to provide a standard for the expression of rules which could easily be exchanged between users and could easily be transmitted across computer networks.

The standard was developed using modular syntax and semantics. Reaction rules comprise integrity constraints (ensuring consistency by triggering an event when something (breaking the integrity rules) happens) and derivation rules which are only triggered when certain conditions are met. An example an integrity constraint could be to check a value is within a range when entered by a user. An example of a derivation could be a simple fact or condition (as a query).

The ontology fragment shown in Figure 9 below shows how a car rental agreement can be represented in RuleML, taken from Boley et al. [143]:

```

<imp>
  <_head>
    <atom>
      <_opr><rel>isAvailable</rel></_opr>
      <var>Car</var>
    </atom>
  </_head>
  <_body>
    <and>
      <atom>
        <_opr><rel>isPresent</rel></_opr>
        <var>Car</var>
      </atom>
      <not>
        <atom>
          <_opr><rel>isAssignedToRentalOrder</rel></_opr>
          <var>Car</var>
        </atom>
      </not>
      <not>
        <atom>
          <_opr><rel>isScheduledForService</rel></_opr>
          <var>Car</var>
        </atom>
      </not>
      <not>
        <atom>
          <_opr><rel>requiresService</rel></_opr>
          <var>Car</var>
        </atom>
      </not>
    </and>
  </_body>
</imp>

```

**Figure 9 - Example of RuleML**

A rental can only be made when a car is available and is present. The car must not be assigned to a current rental. The car must not be scheduled for service and must not require a service.

### 3.7.2 Ontology Rules

RuleML [143] [144] has already been discussed in a previous section, this provides rules in an RDF format. However, as described earlier RDF does not have the expressiveness to describe description logic based semantic description.

Horrocks et al. [144] proposed the Semantic Web Rule Language (SWRL) which combines OWL and RuleML.

An example of a SWRL is shown in Figure 10 below taken from O'Connor et al. [145]

```
hasBrother(?x1,?x2) ^ hasAge(?x1,?age1) ^  
hasAge(?x2,?age2) ^ swrlb:greaterThan(?age2,?age1)  
→  
hasOlderBrother(?x1,?x2)
```

**Figure 10 - Example of a SWRL Rule**

### 3.7.3 Querying Ontologies

SPARQL [83] allows users to specify queries against RDF based semantic description such as OWL. The language has an SQL like syntax, as seen in relational databases.

An example of a SPARQL query is shown in Figure 11 below.

```
SELECT ?driver ?competence ?type ?description WHERE  
{  
  ?driver bm:has_competence ?competence.  
  ?competence a ?class.  
  ?class rdfs:label ?type.  
  ?competence bm:description ?description  
}
```

**Figure 11 - Example SPARQL Query**

### 3.8 Semantic Web Services

Prior to the advent of cloud computing much research was carried out into semantic web services. Semantic information was embedded into web services to aid web service discovery, composition and orchestration. Mark-up languages already discussed such as OWL have been used to describe web services in the same manner web-pages have had semantic mark-up to describe their content.

Domingue et al. [146] Describe IRS-III, which is a broker-based approach to semantic Web-services. It uses a PSM approach to discovering, composing and executing web-services. The broker aspect allows orchestrations of web services to be built.

The approach relies on several languages and ontologies, shown in the list below:

- Web-service Modelling Language (WSML) [147]
- Web-Service Modelling Ontology (WSMO) [148]
- Web Service Execution Language (WSMX) [149]

Much work has been carried out on automatically finding software libraries with a required functionality by researchers such as Gaspari et al. [150], who discuss a competence based matching approach to finding existing software library functionality using a reasoning approach. The semantics of each software library are defined using the Problem Solving Method (PSM) specification syntax Universal Problem-solving Method development Language (UPML), which is an architectural description language specialized for knowledge based systems, as described in the literature review. This approach suffers from a number of issues. Firstly, a specification has to be created for each component, which must be kept up to date with the component as it changes. Secondly, the Problem Solving Method (PSM) syntax is highly mathematical like and may be difficult to

understand and write by an inexperienced user. Thirdly, there is no consideration of if the benefit of re-use is greater than the cost of creating UPML specifications. Lastly, matching is only as good as the specification and searching methodology.

Given the problems of the approach it is still a highly effective at semantic searching and toolsets could be defined to make specification easier. Modern OWL based representations of UPML described by Crubézy and Musen [99] and Dietze (2010) make toolset creation more feasible. Dietze et al. [151] describes finding Web-services/groups of Web-services with a required functionality using a semantic approach which is based on the original approach of Gaspari . The IRS-III broker searches for a set of web-services with a desired competence (functionality) as described by Domingue the candidate web-services are described in terms of WSML and WSMO. The broker does not just match single web-services but can match a group of web-services of a desired competence.

Sheng et al. [152] describe the stages in composition, shown in the list below:

- Definition
- Service Selection
- Deployment
- Execution

Semantic concepts can be used at each stage. In the definition stage a service can be defined using a number of definitions stored as semantic description, as previously stated ontologies based in formats such as RDF can be easily merged.

Service selection can be carried out against semantic description that allows semantically equivalent services to be selected more easily as the semantics of services are clearly defined.

Deployment information for target environments can be semantically described allowing service components to be deployed to semantically defined machine environments.

Execution performance of web-services can be monitored and altered using generic semantic descriptions, masking the low level machine execution environment.

### **3.8.1 Web-Services Description Language**

Bruijn et al. [153] describe the Web-Service Description Language (WSDL), which is an XML based description language describing the services offered by a web-service. Being a “syntactical” description of contracts offered by the service the language does not offer semantic description of the service. It is therefore difficult to compare and select semantically equivalent services. Attempts have been made to add semantic information to WSDL in the same manner WWW web-pages have semantic mark-up.

## **3.9 Ontology Based Descriptions of Service Offerings**

A number of semantic frameworks have been implemented as ontologies, which in turn have been developed using semantic develop tools and mark-ups. Ontologies can be embedded into webpages, as mark-up. Such an approach is seen in the Good Relations ontology [154], which provides semantic information for goods and services.

Ontologies can be standalone and provide support for various knowledge domains. An important domain is the biological sciences and many ontology have been developed. The Evidence Ontology [155] is service for gene sequences.

### **3.9.1 Good Relations**

Hepp [154] has developed the Good Relations ontology which provides a pricing ontology for representing commerce for goods and services between companies. The main motivation for the ontology is to allow consumers to find goods and services from many offerings available in a marketplace, such as internet search. The analogy to cloud computing services can be seen as public clouds allowing users to select processing and storage resources based on price, quality of service and product features. The Good Relations ontology is comprehensive and is used by a number of retailers.

The approach suffers from the fact that it is built as a standalone ontology that considers a single viewpoint, a retail scenario. It is not built from more general description or problem solving ontology that could allow multiple viewpoints to be layered to allow greater usability of the ontology.

### **3.9.2 Evidence Ontology (ECO)**

Chibucos et al. [155] describe Evidence Ontology which allows biological research result evidence to be captured in a controlled and structured manner.

Experimental design and data are stored in a structured along with any academic research published. A “curator” will annotate the academic with terms from ECO, which are structured descriptions of gene sequences. The annotations are added to a gene sequence repository. The gene sequence repository is used to compare gene similarity evidence. The similarities are compared by researchers in the area for gene sequence matches and ECO is update. The ontology is published as a public sequence repository.

The advantage of this approach is description logic based ontology can be used to build semantic linkages between gene sequences in academic research. The ontology provides a structure for gene sequences using

ontology terms which capture semantic linkages between genes and provide a framework for defining new linkages.

### **3.10 Technologies Strongly Related to Cloud Computing**

This section some technologies related to cloud computing are discussed. The advent of mobile technology, mobile computing is being combined with cloud computing to provide edge or fog technology.

Advances in server technology has seen a move to stateless technology seen in Serverless and Lambda computing.

#### **3.10.1 Edge and Fog Computing**

Edge and Fog computing extends cloud computing to combine computing geographically dispersed location aware Internet of Things (IoT) devices, combined with cloud computing services [156] [157]. Devices can be seen as extending the cloud services. The characteristics of the devices on the edge of the network, are a great number of nodes that are location aware and widely geographically distributed, with an interplay between the cloud and the fog/edge. The need for complex semantic description becomes even greater given the greater number of components, relationships and interdependencies.

#### **3.10.2 Serverless and Lambda Computing**

Hendrickson et al. [2] describes Serverless computing as a functional model for computing, moving away from having a number of servers or virtual machines running applications. A stateless development model will be used with applications calling a set of functions that provide application functionality to fulfil user requirements.

The Lambda model of computing is a serverless cloud architecture where developers call a set of handler functions which are managed by the cloud



providers. The handler functions are small and low cost and an application is built from many such functions.

The emerging technologies described will require higher levels of semantic service description as the complexity and number of service components increases.

### **3.11 Semantic Description Development and Toolsets**

This section looks at how ontologies are developed and toolsets that can be used to develop ontologies.

#### **3.11.1 Collaborative Ontology Development**

Walk et al. [158] discuss collaborative ontology engineering projects. The collaborative approach is ideally suited to creating explicit specifications and shared conceptualisations of Cloud Service Agreements (CSA), pricing of cloud services and benefits derived from IS/IT investments from multiple stakeholders. Stakeholders can collaborate using tools such as WebProtégé [159] to work on the structure of the ontology (the terminology or TBox and the relational aspects of ontology or RBox) and the individual instances of the ontology (the assertions or ABox). Such tools allow auditing, change history and correctness of the ontology to be maintained. The process of ontology generation is more difficult than off the shelf collaborative tools that allow Wikis or shared documents be created, as technical help may be required to build a formally correct ontology. The creation of an upper ontology for Benefits Management should provide a template in the form of a complete or semi-complete Terminology Box (TBox) for stakeholders to use.

Sebastian et al. [160] describe an approach to collaborative ontology development using workflows. The researchers highlight the need to define formal workflows for non-ontology experts such as domain experts in the areas of medicine and gene research. This could be extended to business

analysts or those working in the area of Benefits Management. The research outlines a series of tasks that form a workflow for ontology generation, supported by an ontology that describes the process for creating an ontology. This allows those who are unfamiliar with the process of ontology generation to create an ontology from scratch using a collaborative method.

The importance of the change process in ontologies is the subject of the research by Wang et al. [161]. In large scale ontology projects the ability to use and review a change process is part of the ontology building process. Ontology tools such as Protégé [91] and WebProtégé [159] include a change log. The change process is a key factor when a number of collaborators are working on a shared ontology. The ontology engineering process is examined in Strohmaier et al. [162]. The researchers describe four aspects of ontology development is shown in the list below:

- Dynamic
- Social
- Lexical
- Behavioural.

The dynamic aspects of ontology development describe how ontologies change over time. The researchers found that changes occurred in bursts around the project start-up date and, during meetings between collaborators.

The social aspects of ontology development see collaborators working in small groups of two or three people.

The vocabulary of the ontology will stabilise as it becomes mature. This is described as the lexical aspect of the ontology development process and can be measured using a number of mathematical measures of texts such as word similarity or Vector Space Models (VSM) of corpora [163].

The behavioural aspects of ontology development describe how collaborators change the ontology over time. It was found that a change hierarchy saw developers modifying a high level concept and then going on to transform lower level concepts.

Tudorache [159] proposes the usage of WebProtégé as a collaborative ontology editing tool. The tool is light weight in comparison to desktop computer based tools, such as the existing Protégé [91] tool. The WebProtégé tool allows information to be entered via structured input forms which should be familiar to non-technical users, such as domain specialists. The forms can be tailored to a number of user groups. There is support for collaborative working such as threaded discussions, change notifications and change statistics notice boards.

### **3.11.2 Toolsets**

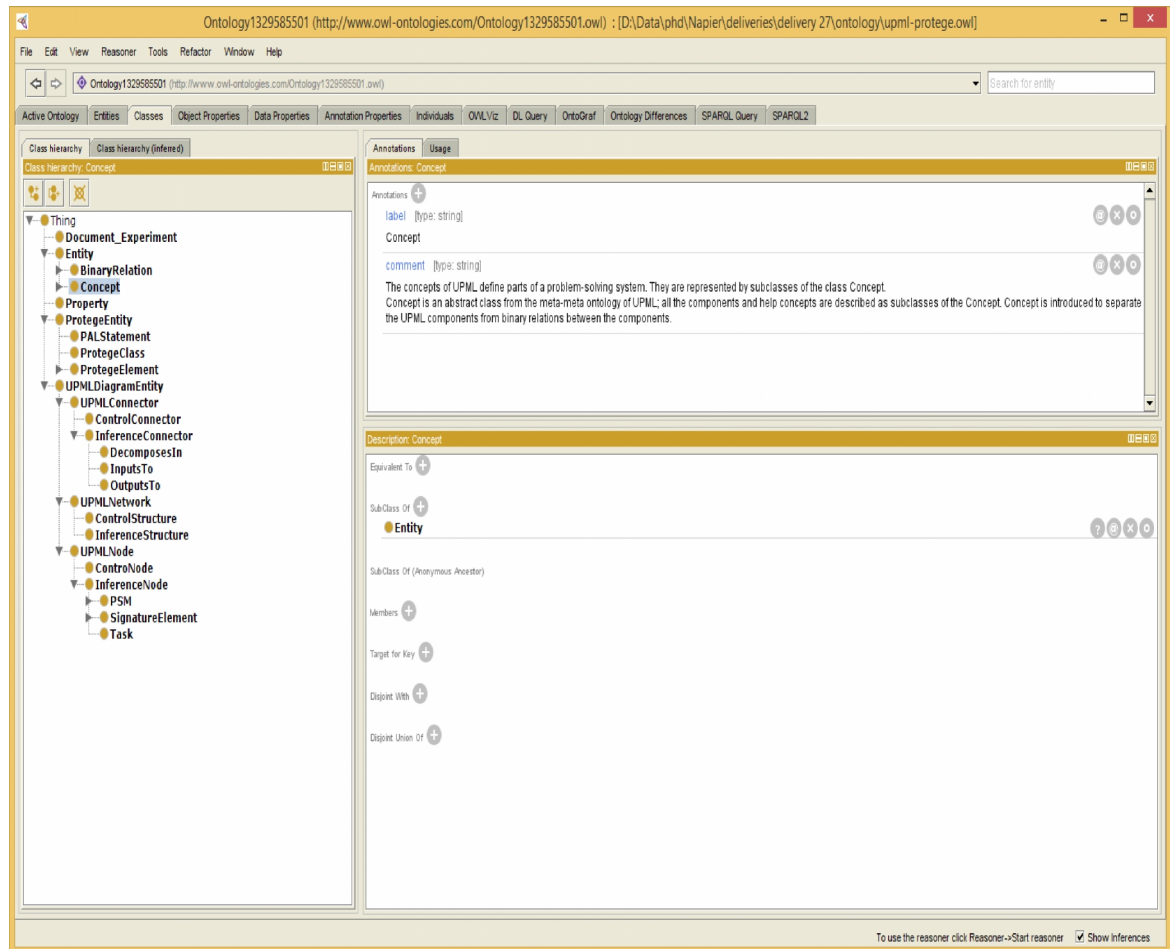
Ontology Editors provide the most important tools in ontology development in languages such as OWL. Alatrish [164] compares a number of ontology editor tools and describes the general features of the tools as the visual representation of TBox, RBox and ABox as tree like structure with class properties and restriction representation. Ability to edit textual descriptions of properties. Ontologies can be overlaid and merged.

Protégé [91] is typical ontology editor for the OWL-2 language, the editor is shown in the Figure 12 below.

A feature of the editor is to offer plug-in extensions for features shown in the list below:

- Fuzzy searching and matching of ontology concepts
- Merging tools that allow auto merging
- Graphing plugins that show network models of ontology

Figure 12 below, shows the Protégé ontology editor



**Figure 12 - Protege Ontology Editor**

### 3.12 Discussion and Conclusions of Related Work

The chapter started with an examination of WWW technologies what have been named Web 2.0, Web 3.0 and Web 4.0. These are arbitrary titles for a collection of technologies, the main thrust of these technologies being cooperation between WWW the use of semantics to build meaning into web-pages. Web 3.0 and Web 4.0 have most relevance to how cloud services are described, used and priced using semantics. The discussion moved on to technologies are used to achieve semantic description of web-pages.

HTML based semantic mark-up such as Microdata and Microformats provide a simple way to embed limited semantics into web-pages without having to learn new languages and concepts. However, this approach is flawed as HTML is primarily a presentation technology and the description and linkage capabilities offered by the technologies is limited.

Full semantic description mark-up is offered by RDF based mark-ups such as OWL. These mark-ups offer full semantic description, with the ability to support all aspects of description logics. There are query languages and rule languages available. The issues with these mark-ups is they use verbose XML descriptions that must be held as separate mark-ups from the HTML for a web-page. Web-developers must learn new concepts such as ontologies and use new tools, and thus there is a large learning curve for users.

JSON-LD is a WWW technology, like HTML based mark-ups, and is more familiar to WWW software developers compared to RDF based mark-ups. JSON based mark-ups are less mature than RDF mark-ups but are more familiar to developers and are less verbose than XML based mark-ups.

Triples databases offer the ability to implement ontologies robust commercial manner, fast processing of queries in languages such as SPARQL make complex semantic processing in real-time possible. This is important in cloud service selection and composition, which may require a large number of service component combinations to be considered in real-time.

Semantic web-services are the closest technology to semantic cloud services and therefore the most important source for related work. Much work has been carried out on semantic web-service mark-up using RDF based technologies. This work is directly applicable to cloud service discovery, composition and orchestration. Cloud services tend to be more commercialised, and therefore service agreements and contracts and pricing are more applicable to cloud services.

There have been a number of successful ontologies developed in areas such as commerce and biology. The Good Relations ontology is used as embedded ontology in a number of commercial web-sites, this allows users to find products they require more easily. Biological ontologies such as ECO guide users and help them classify research more easily.

There are emerging technologies in mobile and server based applications that make semantic concepts even more important and relevant. Fog and edge computing will increase the complexity of service discovery, composition and pricing due to the cloud/mobile interface and the number of service combinations possible. Lambda computing will also increase the complexity of service composition as a service description will be built from many small functions which must be combined.

Toolsets and supporting infrastructure for ontology development are important when selecting a mark-up or technology to develop semantic description. The most mature toolsets are in RDF based technologies.

Development of large ontology, such as those required by cloud service description for Cloud Service Agreements, requirements, pricing and benefits management requires collaboration between groups of researchers, domain experts and ontology developers. The dynamic. Social, lexical and behavioural factors required for collaborative ontology development have been discussed. Emerging collaborative ontology development toolsets, such as WebProtégé are key to supporting collaborative ontology development.

## **Chapter 4      Overview on the Proposed Approach**

### **4.1 Chapter Overview**

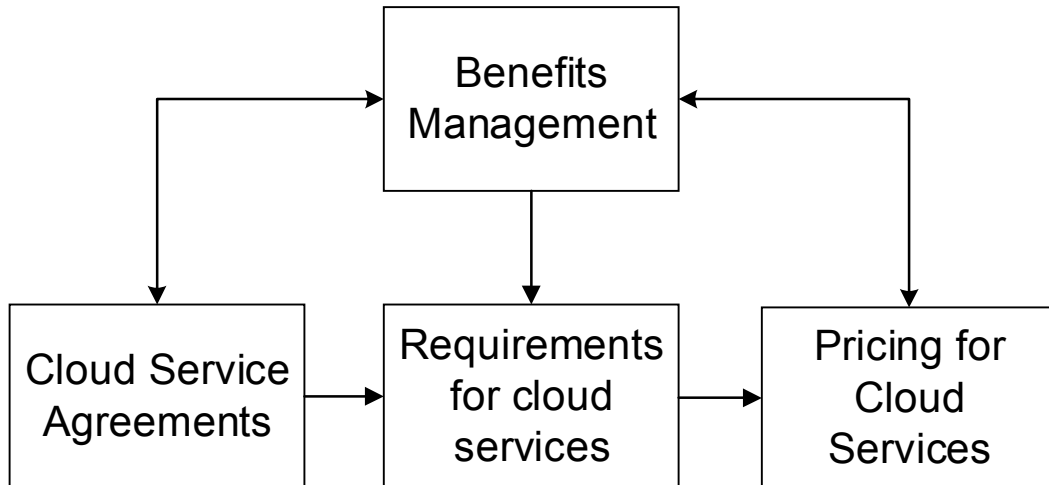
The proposed framework is introduced in this chapter as the main contribution to knowledge. The contributions to the four elements of the framework (agreements, requirements, pricing and benefits management) are described along with the contribution to their combination into the proposed framework.

### **4.2 Introduction**

The main contribution to knowledge provided by the research is a unique framework that brings together elements of cloud service description using a common semantic description. The framework uses a common semantic representation to model various aspects of cloud computing using Problem Solving Ontology (PSO). This allows cloud computing requirements to be modelled as Tasks, Problem Solving Methods (PSM) and Knowledge Domains. This approach allows a common language of discourse between cloud users and cloud providers to generate a “utility market”.

The usage of generic Problem Solving Methods and Knowledge Domains leads to greater reuse of framework information.

A theme running through research into “An Approach to the Semantic Intelligence Cloud” is to build semantic description for aspects of cloud computing from a starting point of agreement through to pricing with consideration of the benefits delivered throughout the process, as showing in the Figure 13 below.



**Figure 13 - Overview of Semantic Framework**

The proposed approach builds on the theory described in Chapters 2 and 3, the literature review and related work respectively, to synthesise important ontology for cloud computing in a number of areas that are sparsely covered by current research into cloud computing service description using semantics. In cloud service agreements researchers have concentrated on Service Level Agreements (SLA), Chapter 5 attempts to provide ontology for all aspects of cloud service agreement as defined by the Cloud Standards Customer Council [6].

When a customer and service supplier has made a service agreement the next stage is to define cloud service requirements. Chapter 6 provides ontology for cloud service requirements. High level user requirements are mapped to low level service specifications using semantic technologies.

Requirements expressed as ontology provide the basis for pricing decisions. Chapter 7 provides a considerable contribution to pricing cloud services by providing semantic framework for ontology for pricing of cloud services.

Encompassing the areas of cloud computing service agreements, requirements and pricing, Chapter 8 provides a unique study into a semantic framework implemented as ontology for management of benefits produced



by cloud computing investments. This research builds on many years of previous research into benefits management of cloud computing investment to produce a unique semantic framework to consider benefits generated by cloud computing investments.

The chapter will continue with a detailed consideration of Cloud Service Agreements, requirements, pricing and benefits and will be completed with a discussion and conclusions.

### **4.3 Contribution of the Proposed Framework**

The proposed framework brings together Cloud Service Agreements, requirements, pricing and Benefits Management into a single semantic framework based on semantic representation. Concepts and relationships from each area can be mapped and traced from agreements to pricing with Benefits Management as a theme running through each areas. The end goal is to provide a Unified Semantic Framework for Cloud Service Description.

Previous frameworks have concentrated on a single area such as Service Level Agreements or pricing infrastructure. The approaches have mainly used 'syntactical' approaches, mapping and tracing at a superficial level.

The chapter will now continue with contributions for each of the four areas in the proposed semantic framework.

### **4.4 Contribution to the Semantic Description of Cloud Service Agreements**

The Cloud Standards Customer Council (CSCC) have produced a large amount of documentation on Cloud Service Agreements (CSA), which comprise Customer Agreements (CA) Service Level Agreements (SLA) and Acceptable Use Policy (AUP). The contribution in this area is to produce

semantic description framework based on this documentation and the work of other researchers in this area.

Minimal research has been carried out in this area, it is mainly “Syntactical” in nature, expressing concepts and relationships in a simplistic manner. The bulk of research has been into SLA, leaving CA and AUP which very little, if any research.

The semantic analysis of CSA provides a unique contribution to research providing more meaningful models compared to simple “syntactical” or mathematical models currently being used in research.

Developing semantic description allows the documentation and research in this area to be analysed, searched and reasoned on, allowing researchers and practitioners in this area to utilise a unique resource to develop further semantic description and toolsets to analyse CSA, shown in the list below:

- Merge and add legal semantic description for the contractual aspects of CSA
- Develop tools to guide the CSA creation process
- To analyse CSA using the semantic description
- Create automated analysis of CSA across multiple cloud providers

Legal aspects of CSA are discussed in Chapter 5, giving the unique ability to relate agreements to legal and contractual semantic description.

The comprehensive semantic description developed provides other researchers with a resource to develop toolsets to guide users through the agreement process, for example, given a set of user requirements, how the requirements can be fulfilled by a Customer Agreement.

Existing agreements can be analysed against the terminology (TBox) in the semantic description developed, allowing comparison of agreements using

semantics and, to highlight any issues in the agreement, for example missing elements. Semantics allow terminology differences in agreements to be managed via common terminology found in the TBox. Analysis at a semantic level allows strategies, such as pricing strategies, to be developed more easily across a number of cloud providers, as service characteristics that are equivalent can be compared.

The knowledge of multiple groups can be combined into the semantic description developed, for example pricing researchers and legal researchers. Automated decision making tools can be developed to assess CSA.

## **4.5 Contribution to Cloud Service Requirements**

Chapter 6 provides contribution to knowledge in cloud service requirements in two areas, requirements as semantic description and usage of a specialised Problem Solving Ontology (PSO).

Semantic description is used to specify cloud service requirements in a precise manner that can be reused and merged easily with other ontology. PSO based approaches are developed for cloud service requirements.

The PSO approach applied to cloud service requirements breaking requirements into tasks that use generic problem solving methods, which can be applied to a number of knowledge domains (for example bioinformatics or manufacturing).

A new cloud service semantic description design was synthesised, building on the Unified Problem Development Method Language (UPML) represented as OWL-2 ontology, the new semantic description was overlaid on the UPML ontology using its base constructs to express newly developed ontology.

The benefits provided by the framework centres around precise and unambiguous description that can be reasoned across to create new knowledge and can be merged with other semantic descriptions of cloud services. The ability to map requirements to Cloud Service Agreements (CSA) and other ontologies, for example legal and contract ontologies for cloud computing, with the ability to map such ontologies across a number of legal jurisdictions for example the common law jurisdictions (USA, UK, Hong Kong and Australia), civil systems (mainland Europe and China) and Hybrid of the civil and common law (Japan).

The identification of key brokerage elements are high level and low level requirements below. The brokerage (of requirements) process describes how user requirements are described, so that they can be mapped to CSA, pricing and low level requirements. The unique contribution in this area is to identify these requirement and demonstrate mapping at a theoretical level and as case studies.

A key contribution is to relate business benefits to requirements in cloud computing services. Very little research has been carried out in this area. Using semantic techniques it was possible to relate concepts from Benefits Management techniques to high level requirements for cloud computing investments.

The move from semantic web-services to cloud computing services has seen the rise in the importance of cloud service pricing. Cloud services are commonly offered on public clouds, whereas semantic web-services tended to be in-house for use within the organisation that built the web-service. The unique contribution from the research carried out is to provide a semantic description of pricing of requirements related to CSA, Benefits Management and the in-depth research into cloud computing service pricing seen in later chapters.

Cloud services are not static and information from cloud services needs to be fed back into low level requirements and low level requirements into high level requirements, increasingly this is required in real-time as on-demand and spot resources are supplied and consumed.

#### **4.5.1 High Level Requirements**

The new semantic framework provides high level requirements (Brokerage) and low level requirements (interface and abstractions of physical cloud services).

Discovery requirements for finding suitable service or components:

Mediation requirements are used to resolve mismatches between user requirements and features and competencies offered by service providers. Requirements descriptions are matched to service descriptions using merging and matching of semantic descriptions at concept, relationship and assertion levels.

Choreography requirements deal with the organisation, interfaces and exchanges between cloud services to achieve the functionality required by users. Each cloud service will be self-describing. The semantic framework provides enhanced choreography description.

Requirements for adaption are concerned with the manner in which cloud services can be adapted given a set of user requirements. Cloud services can be adapted in a number of ways. The framework developed for semantic requirements is used to map user requirements to low level requirements to drive adaptation of requirements. Firstly, the service can be physically changed to meet user requirements exactly. Secondly, adaption patterns can be used to provide a façade to a provider's service adapting the service by supplying service additions or changes in functionality behaviours or combining it with other services.

Grounding requirements can be described semantically using the developed framework. The grounding is seen as a mapping between the layers of the semantic requirements frameworks. Firstly, between high and low requirements. Secondly, between low level requirements and the cloud service providers' service description. The semantic requirements framework provides a contribution at three levels. Concepts such as generic Problem Solving Methods (PSM) can be mapped at three levels. In high level terms a description of the need for a routing requirement, as a low level requirement level a travelling salesman PSM and at provider's implementation level a virtual machine resource and program to provide the functionality.

Requirements for monitoring are seen at two levels. Firstly, at a high level a user may specify requirements from Cloud Service Agreements (CSA) discussed previously. The semantic framework provides a unique contribution in this area by providing terminology that can be mapped between CSA and requirements. Secondly, low level requirements will provide detailed and absolute targets for monitoring and provide mappings to high level requirements and to terminology for physical cloud services. The semantic framework provides a unique terminology and mapping mechanism for relating monitoring requirements to CSA.

The semantic framework provides precise and unambiguous specifications of cloud services that can be reasoned across. It is possible to compare semantically equivalent service specifications at a high level, even if the services are specified by providers very differently at a 'syntactical' level.

Fault handling describes the concepts, relationships and assertions of action when a service threshold is breached. These requirements are closely related to service specification requirements seen in Service Level Agreements (SLA) and Acceptable use Policy (AUP) previously discussed in CSA. A unique contribution is to allow relationships to be established between SLA and AUP and fault handling concepts.

Pricing requirements have become increasingly important as cloud service providers have made public offerings of services, platforms and infrastructure. The semantic framework described in this thesis not only describes pricing requirements as high level requirements, but also provides a unique contribution in relating low level cloud service pricing description, pricing in CSA and focuses on detailed semantic mechanisms in later chapters.

#### **4.5.2 Low level Requirements**

Low level requirements consider the mechanism that maps user requirements described in high level requirements to cloud service providers' service specifications.

Resource description uses semantic descriptions to model and abstract the low level resources provided by cloud computing services, for example CPU capability, storage capacity and network characteristics.

Suppliers of public cloud services provide complex and comprehensive pricing information based on the resource descriptions described previously. Many of the services supplied by cloud service providers are semantically equivalent, but described in a number syntactically different forms. A major contribution is semantically describe cloud pricing structures.

Cloud interfaces adapters and bridges requirements describe how high level requirements will be mapped to physical cloud services.

### **4.6 Contribution to Cloud Service Pricing**

The major contribution from this research is a complete workflow for providing pricing information for cloud computing resources based on semantic description. This builds on previous work on cloud computing requirements engineering, cloud service description and the semantic

description of cloud pricing concepts. This provides a progression from simple syntactical and mathematical techniques provided by current approaches.

A unique multi-layered semantic description built on UPML was developed, allowing a problem solving task driven approach to be developed. Tasks were built using generic problem solving methods which work against knowledge domains specific to the business or universe of discourse. Requirements are mapped to low level pricing information using tasks.

The approach is superior to existing pricing approaches for the following reasons, shown in the list below:

- The semantic nature of the description can be used to represent complex pricing relationships
- Semantic framework concepts and relationships are used to abstract differences in individual vendors pricing structures
- Semantic framework terminology can easily change from information fed back from cloud services
- Generic problem solving methods can be instrumented for their notional cost and reused across a number of knowledge domains

## **4.7 Contribution to Benefits Management in Cloud Computing**

Most research into the economic and non-economic benefits derived from cloud computing has focussed on generic business discussion research or found in business articles and in magazines. There has been substantial research into economic benefits derived from cloud computing at a macro or micro economic levels.

A major contribution of the research contained in this thesis is to use a semantic framework to analyse a number of cloud computing case studies



derived from primary and secondary case studies. An existing approach used successfully for analysing benefits derived from information systems implementations 'Benefits Management' [4] was used.

Firstly, the Benefits Management approach was represented as a semantic framework, enhancing the current approach. The semantic framework was then used to analyse the case studies, 'encoding' the case studies into concepts, relationships and assertions. The analysis allowed generic concepts and reasoning to be extracted from the research that can be applied to Benefits Management of cloud computing investments. A knowledge base for other researchers to use. The semantic enhancement of the Benefits Management method could also be used by other researchers.

## **4.8 Discussion and Conclusion**

The unique contribution knowledge comprises a proposed semantic framework which brings together four definitions which are based on description logic and the specialisation of problem solving ontology. The semantic descriptions provide semantic description of cloud services that are important to the specification of cloud services but are not well represented in research literature. There is extensive use of problem solving ontology [165], which specifies requirements as tasks, this is well suited to cloud service specification, especially new serverless approaches.

The semantic description can be merged to form a meta-semantic description for specification of cloud services, this can also be connected to other semantic descriptions, for example for legal cloud services. The development semantic description framework also considered collaborative semantic description development tools and, the semantic description developed in this thesis has been deployed to a collaborative semantic description platform for usage by other researchers.

Cloud Service Agreements (CSA) are essential to cloud users as they represent a “contract” between the customer and the provider. The CSA will contain many terms and conditions that a user must read and understand and which will have major operational financial implications. Research has focused on the operational aspects of Service Level Agreements (SLA), for example average start time for a virtual machine. This is a very narrow focus, which excludes many important aspects of CSA and many researchers have developed only simple “syntactical” models for SLA. The Customer Agreement (CA) and Acceptable Use Policy (AUP) are very important aspects of CSA. Representation as semantic description provides a number of benefits, the development of common terminology and relationships that allows CSA from providers to be compared. Services can be developed to compare many CSA in real-time, so that cloud services can be selected in against changing public cloud resource markets, which is important for all types of cloud instances, pre-purchased (reserved) instances can be analysed to provide the correct mix of services, on-demand and spot instances can be selected in real-time.

The research undertaken provides a unique multi-layered semantic description for requirements built on UPML. This distinct task orientated approach sees requirements built as a series of tasks, which utilise a series of generic problem solving methods working against a knowledge domain specific to the problems being solved [166]. The approach sees high level requirements expressed as problem solving semantic description, which are mapped onto low level semantic description which abstracts the concepts seen in physical cloud services using semantic constructs. This approach provides a number of advantages over existing “syntactical” or mathematical techniques, providing abstraction of cloud services which can be reasoned on to generate new knowledge, without the need to specify every physical cloud service in detail. The task based approach provides a framework for service specification, with easy mapping between high and low level requirements due to common representations of terminology and assertions.

The majority of research into cloud computing pricing utilises syntactical or mathematical models. The simplistic models have difficulty in expressing the complexity of relating requirements to complex pricing structures seen in cloud computing services. A semantic description framework approach allows generic pricing constructs to be represented hiding the detail of individual cloud providers' pricing structures. Again, the unique application of problem solving frameworks to pricing allows the mapping of high and low level requirements from research into cloud service requirements to pricing structures.

A unique application of the Benefits Management approach to cloud computing has been presented. Further novelty is seen in the representation of the Benefits management approach as a semantic framework, with a large selection of case studies obtained from primary and secondary sources.

The usage of semantics to represent Benefits Management again allows benefits from cloud computing investments to be related to Cloud Service Agreements, requirements and pricing.

In conclusion, the common representation of cloud computing service terminology as semantic description allows a common representation to be used for all aspects of cloud computing services and, allows the research to be extended to include work from other researchers. The problem solving approach allows the task to be used as a unit of work and leads to greater reuse by usage of generic problem solving methods.

## **Chapter 5      Semantic      Description      for      Cloud Service Agreements**

### **5.1 Chapter Overview**

Cloud Service Agreements are the starting point for cloud service requirements and the relationship between a cloud service provider and a cloud service user. It is important to codify these agreements in a semantic representation which captures the complexity and meaning in the agreements.

Cloud Service Agreements comprise Customer Agreements, Service Level Agreements and Acceptable Use Policies, with each described in detail.

A case study provides semantic description of real Cloud Service Agreements and synthesises a possible generic semantic description for agreements.

### **5.2 Introduction**

Description Logics (DL) allow the syntax and semantics of CSA to be modelled explicitly, formally and precisely. Tools such as Protégé are used to build a semantic framework implemented as description logic based ontology. This allows researchers and developers to test the correctness and reasoning across the models.

Axioms are statements that form DL based ontologies. DL axioms can be categorized into three types, a Terminology Box (TBox), a Relational Box (RBox) and an Assertion-Box (ABox). The TBox defines terminology of concepts, for example in cloud security is shown in Figure 14 below:

“A vulnerability is exploited by a security threat”.

This could be represented as:

“Vulnerability is\_exploited\_by Threat”

#### **Figure 14 - Cloud Security Concepts**

Once formalised, a vulnerability could be defined as infrastructure, data, access or regulation based. A threat is defined as being technically or process driven. The terminology box allows concepts such as inclusion, inheritance and negation to be represented by axioms and for reasoning to take place across axioms.

The RBox axioms describe the properties of roles, for example role-inclusion and role-equivalence. An example seen in cloud security is shown in Figure 15 below.

Authorized\_user is\_a\_subrole\_of User

User is\_a\_subrole\_of Person

From this it can be inferred that an “Authorized\_user” is a person, which is an example of role inclusion.

A role equivalence example from cloud security could be:

Has\_Technical\_Threat\_ Is\_equivalent\_to Has\_Threat AND type(Technical)

#### **Figure 15 - RBox Axioms for Security**

The ABox describes knowledge about named individuals. An example from cloud security are shown in Figure 16 below:

is_exploited_by(Vulnerability: Access, Threat: gets_between)
gets_between(Person: John, System: Management_front_end)
Person(John)
System(Management_front_end)

**Figure 16 - ABox for Security**

It can be asserted that 'John' is a person and 'Management front end' is a system.

Semantic description can be queried by languages such as SPARQL [133], which allow information to be extracted from a semantic description, facilitating the generation of sophisticated reports. Software packages can be built around ontologies. New knowledge can be created by reasoning.

It has been demonstrated that a DL based semantic description can precisely define terminology, relationships and assertions for aspects of CSA. A number of detailed examples are shown in the next section.

### **5.3 Semantic Description for Cloud Service Agreements**

A number of examples of how ontologies can be developed from work on CSA will be given. The examples provided show semantic description for Customer Agreements, Service Level Agreements and Acceptable Use Policy.

### **5.4 Customer Agreements**

The usage and payments aspects of CA deal with the transformation of user requirements into conceptual resource usage. The model for usage and payments is shown in Figure 17 (below). Requirements can be defined as tasks that combine process descriptions or algorithms and data models to fulfil user requirements. Processing usage concerns the amount of compute

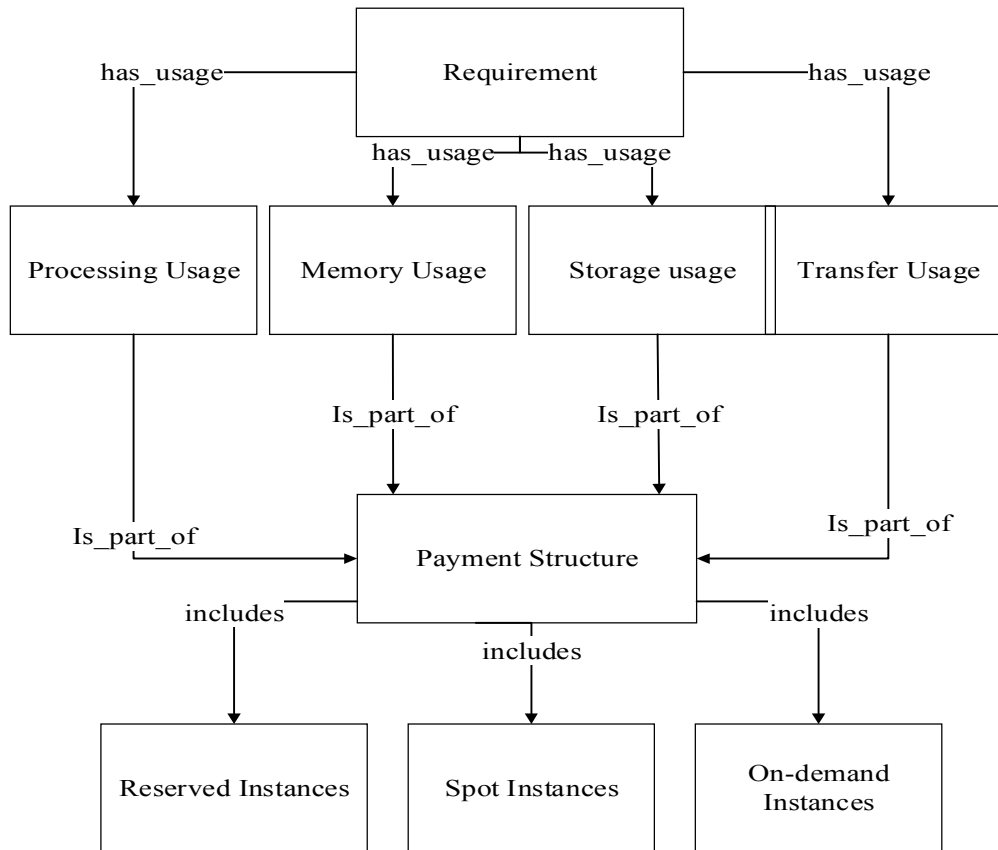
resources needed to complete a requirements task. It is possible to calibrate conceptual tasks to processing usage models. Lower processing usage may be traded-off for longer processing time at lower cost.

Memory usage defines the conceptual average and peak memory usage required to fulfil a requirements task. Users may again select memory usage models for their cost, quality and time requirements.

Storage usage is the conceptual size and storage model (such as relational database, NoSQL databases or simple storage) used to hold data models for user requirements. The storage usage describes backup strategies, geographical location and data transfer quality of service.

Transfer usage occurs when data is transferred to and from cloud services, such as the user uploading data for processing or downloading results from processing. Data can also be transferred between geographical data centres to improve processing performance or for legal reasons.

Each usage description maps to a payment structure that is built of cloud instance profiles. The instances comprise reserved instances that provide lower cost pre-purchased resources, spot instances are purchased on auction markets, and on-demand instances purchased on a pay-as-you-go basis, which may be more expensive.



**Figure 17 - Usage and Payment Structure TBox**

An example of the assertions (ABox) for usage and payment is shown in Figure 18 below.

```

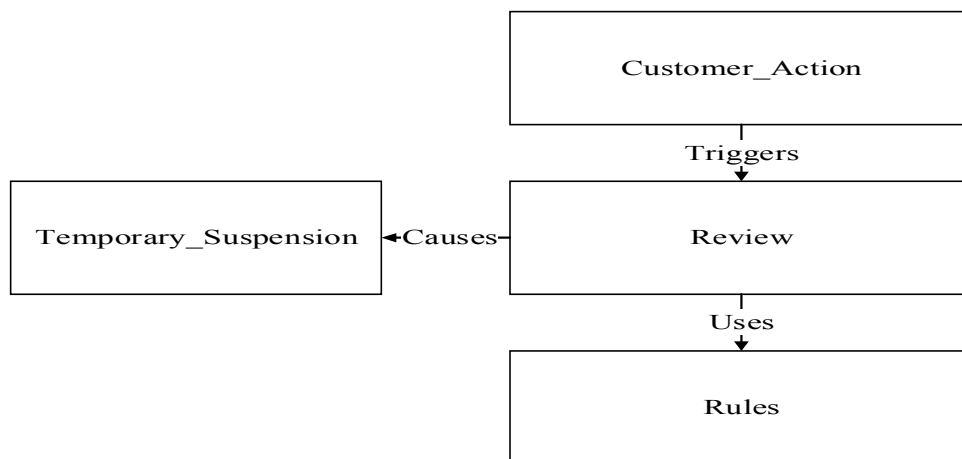
Requirement (Calculate Forward Price of Financial Derivative for 10,000
instruments)
Processing_Usage(0.2 Hours for Small Instance)
Processing_Usage(0.1 Hours for Large Instance)
Memory_Usage(Average:0.8GB, Peak 1.2GB)
Storage_Usage(Simple:8.8GB)
Transfer (In: 6.5GB, Out 2.3GB)

PaymentStructure_SlowAction(Reserved_Instances(2:Small:$10))+Transfer
Cost($8)
PaymentStructure_Express(Reserved_Instances(2:Large:$40)+(1:Spot:$4.5
0)+(1:On_demand:$6.00)) +TransferCost($8)
  
```

**Figure 18 - Usage and Payment ABox**



The temporary suspension semantic description terminology is shown in the Figure 19 below. Customer actions such as sending bulk e-mail or hosting illegal material is reviewed using rules defined in the CA. The review process may lead to temporary suspension of all or some services for a specified period.



**Figure 19 - Temporary Suspension of Agreement TBox**

An example for a temporary suspension ABox is shown in Figure 20 below.

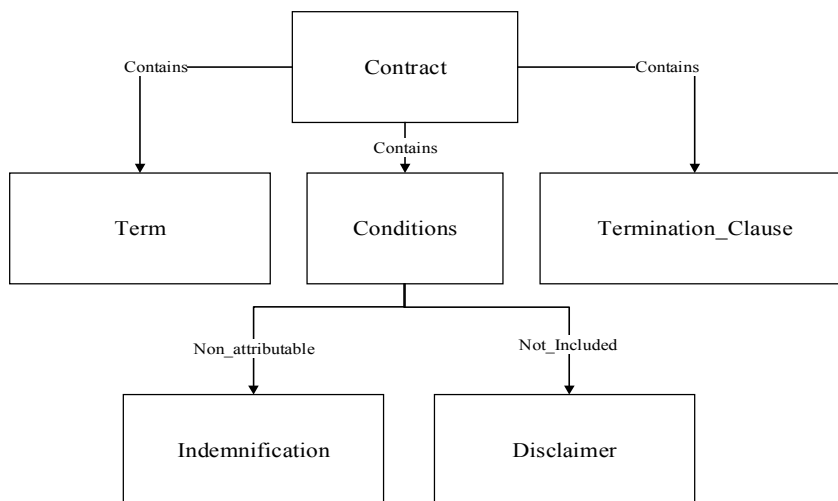
Customer_Action(Hosting_illegal_material) Review(Examine_Material) Review(Remove_Material) Rule(Suspend_account_for_10_days_on_first_breach) Rule(Suspend_account_for_20_days_on_first_breach) Temporary_Suspension(deny_all_access)
---

**Figure 20 - Temporary Suspension ABox**

The terminology for the legal aspects of CSA semantic description is shown in the Figure 21 below. A contract is formed between customers and providers. The contract contains a number of terms that describe the features of the contract. Terms can be express and, are explicitly stated in the contract, an example of an express term could be the prices for cloud services.

Implied contract terms are defined by custom or working practices between the parties in the contract, an example of an implied term could be data privacy in a given jurisdiction.

Conditions are a promises parties in a contract must meet, failure to do so will result in a breach of contract, which may result in penalties or termination of the contract. Conditions can be split into indemnification and disclaimers. Indemnification describes compensation for losses resulting from usage of the service. Disclaimers outline what is not included in the contract. Termination clauses allow either party to end a contract by providing written notice.



**Figure 21 - Legal Aspects of CSA Semantic Description**

The Abox for legal aspects of CSA is shown in Figure 22 below.

```

Contract(Terms AND Conditions AND Termination_Clause)
Conditions(Condition OR Indemnification OR Disclaimer)
Contract(Contract_for_cloud_Infrastructure_services)
Term(Price_per_Hour($7.50))
Term(Location(European_Union))
Term(Period(12 Months))
Condition(Monitoring(Sample_Pattern_of_Usage))
  
```

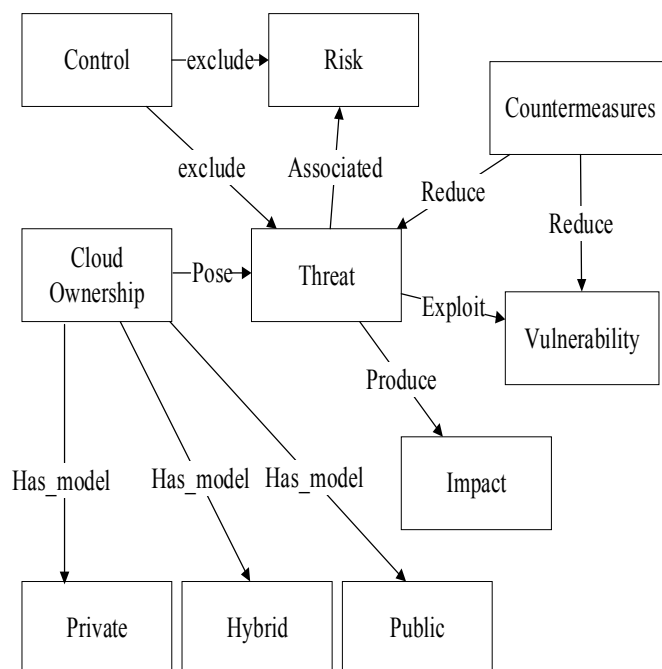
```

Condiiton(Law(EU_Law))
Indemnification(Loss_of_Service(Provide_Credit))
Disclaimer(Cap(Direct_Loss($100,000)))
Termination_Clause(Trigger(Illegal_Usage))
Termination_Clause(Data_Removal(Months(3)))

```

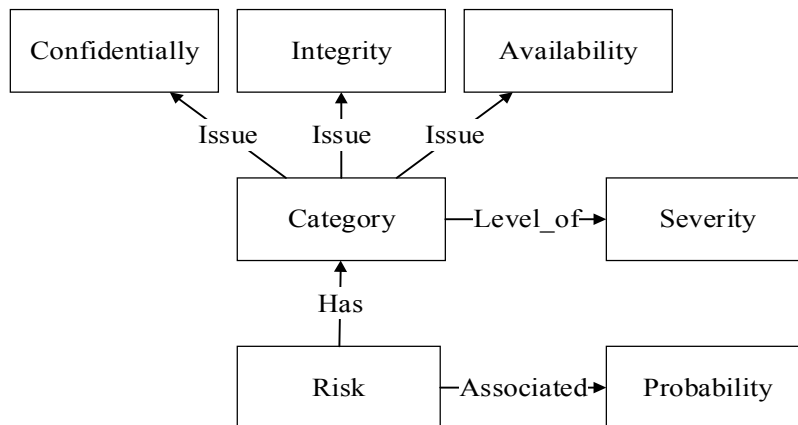
**Figure 22 - Legal Aspects of CSA ABox**

High level security terminology for cloud service agreements associate cloud ownership models such as private, public and Hybrid cloud ownership with a number of threats which exploit vulnerabilities in cloud security systems. Threats produce impacts which can be measured on a defined numeric scale. Risk levels can be calculated by considering risk severity and the probability of a threat occurring. An overview of the security terminology is shown in the Figure 23 below.



**Figure 23 - Security TBox**

Risk has a number of categories which are confidentiality, integrity or availability. Each risk category has a level of severity associated with it. Risks also have a probability associated with them. The terminology for this is shown in the Figure 24 below.



**Figure 24 - Risk TBox**

The terminology combined with impact from the security terminology defined previously is used to calculate a risk score, which is an absolute measure of risk, an example of which is shown in Table 6 (below).

Impact is rated on a scale from 1-10 with and Severity comprising three components of Confidentiality, Integrity and Availability each having a scale from 1-10 are summed to provide an overall Severity score. A probability is defined on a scale from 0-0.9. Impact, severity and probability are multiplied to provide an overall absolute risk score which can be compared to other absolute risk scores, as a relative measure, to form a pattern of overall absolute risk in a collection of cloud computing resources.

Table 6 (below) shows some example figures. In operational ontology for Customer Agreements the customers and cloud service vendors will define initial figures for each threat. The figures can be adjusted in an operational context over time, as the cloud computing resources are utilised.

Threat	Impact	Category	Severity	Probability
Data Leak	2	Confidentiality	8	0.1
		Integrity	8	
		Availability	2	

**Table 6 - Risk Probability Score Calculation**

An example of the risk calculation is shown in Figure 25 below.

Risk score = Impact\* SUM(Severity Components)\* Probability

$$3.6 = 2 * (8+8+2) * 0.1$$

**Figure 25 - Example Risk Score Calculation**

The Risk Probability can be calculated using a SPARQL query. The assertions (ABox) used were developed from the terminology, some examples are shown in Figure 26 below.

Threat(Data\_Leak)

Vulnerability(Access)

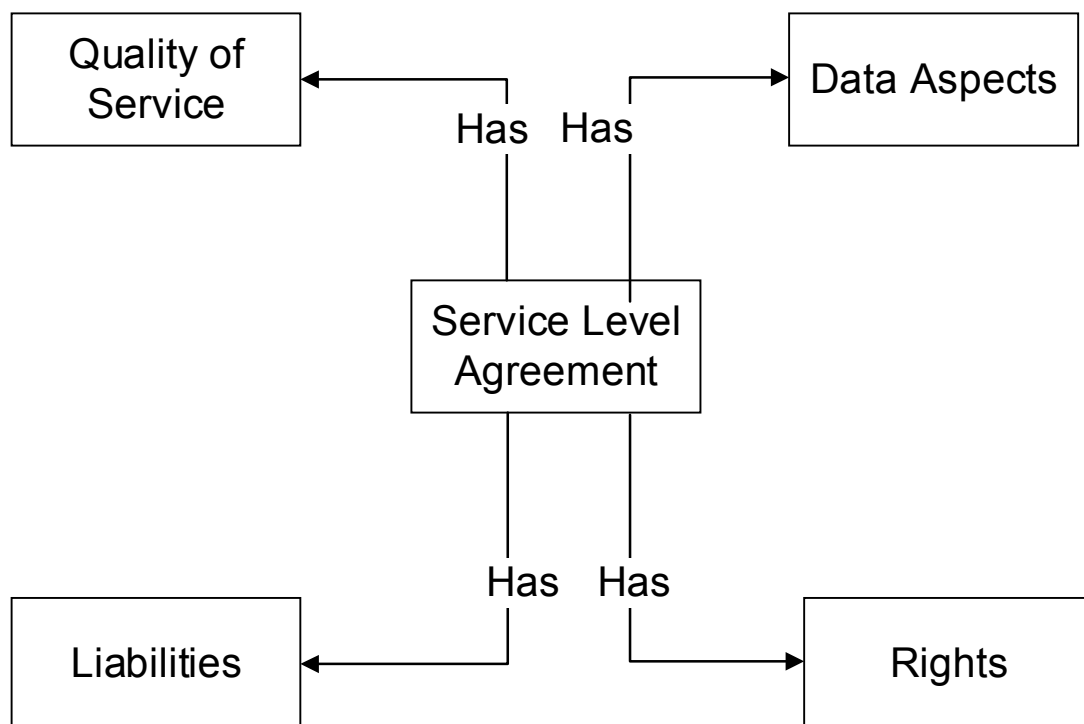
Impact(2)

Risk(Loss\_Of\_Intellectual\_Property)

**Figure 26 - Risk Probability ABox**

## 5.5 Service Level Agreements

Figure 27 shows the high level TBox and RBox for Service Level Agreements. Quality of service deals with issues such as service response time, reliability and availability. Data aspects concern data integrity, data preservation, and data disclosure and data location. Rights are rights over the service, proprietary rights and warranty. Liability of the cloud service provider can be direct or indirect and has limits.

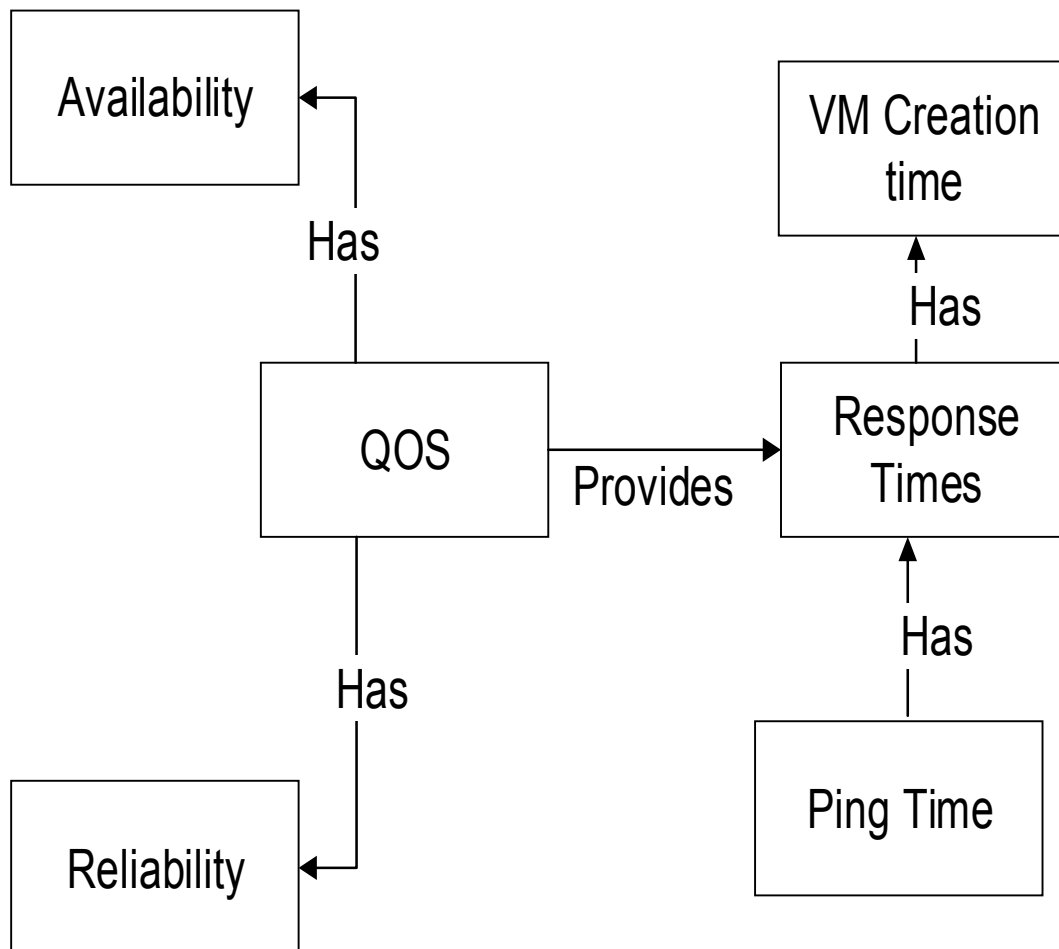


**Figure 27 - SLA Terminology and Relations**

Quality of Service (QOS) has a number of possible criteria, such as response times, availability and reliability of cloud services, shown in Figure 28 below. Availability relates to the percentage of time a cloud service is available and this is usually close to 100%.

Reliability relates to the number of failures in a given period and Mean Time Between Failure (MTBF).

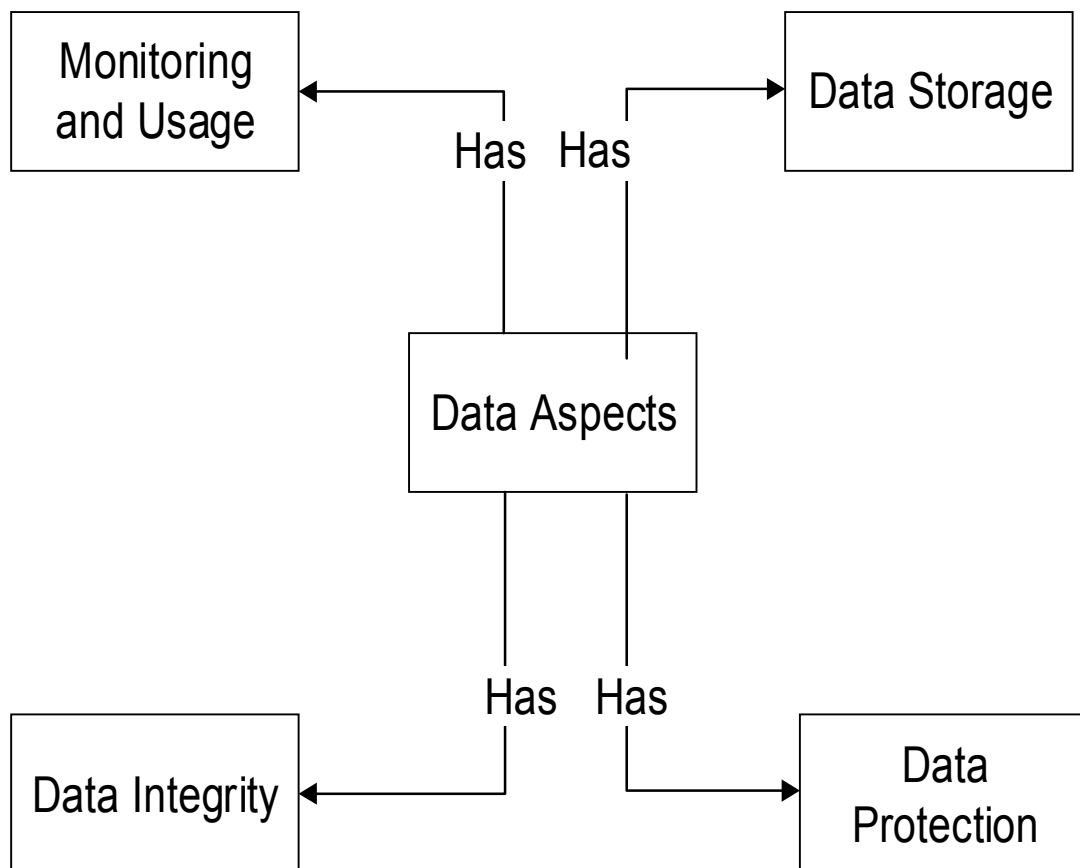
Response times are measured by criteria such as the time to create a new virtual machine and ping times which will be dependent on the geographical local of the cloud service.



**Figure 28 - Quality of Services Terminology and Relations**

Data aspects deal with Data storage and protection which are critical issues in the SLA, shown in Figure 29 below, with many suppliers stating their policies as express terms in contracts. Many suppliers state that data integrity is the responsibility of the customer. A small number of suppliers provide some assurances towards data integrity. On termination of an agreement some suppliers will delete data immediately, others will allow a grace period so that data can be downloaded by customers. Monitoring of services is not covered by some SLA. A number of service providers will

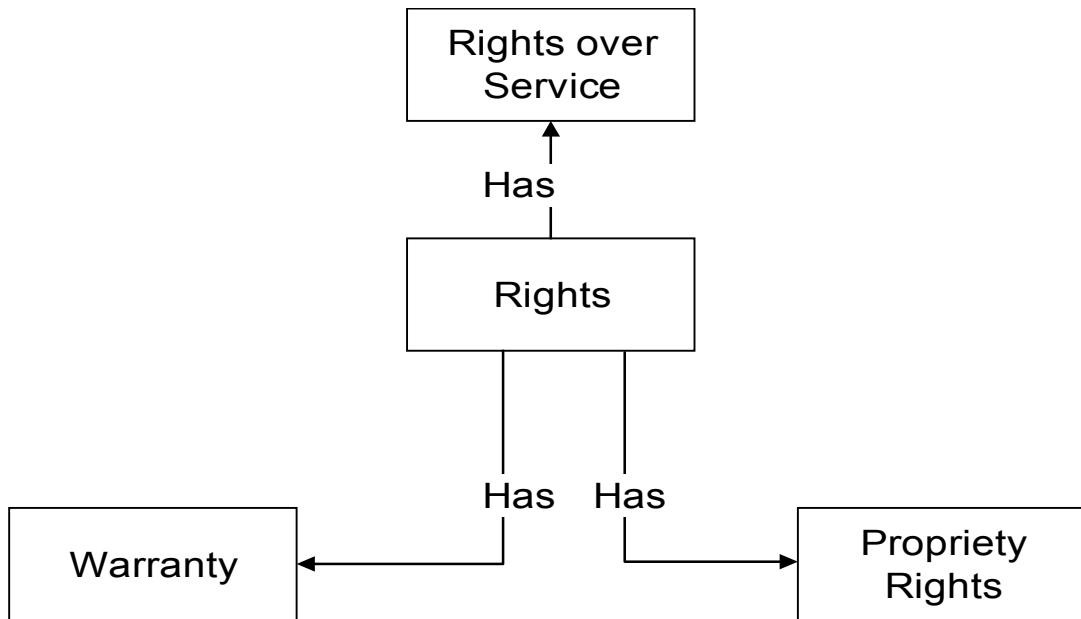
monitor usage patterns and some will monitor actual service and data usage. Transfer time and security of data. The amount of data stored may influence the price a customer pays for cloud services. Transfer time is strongly linked to the co-location of data and data processing units in the same geographic location.



**Figure 29 - Data Aspects Terminology and Relations**

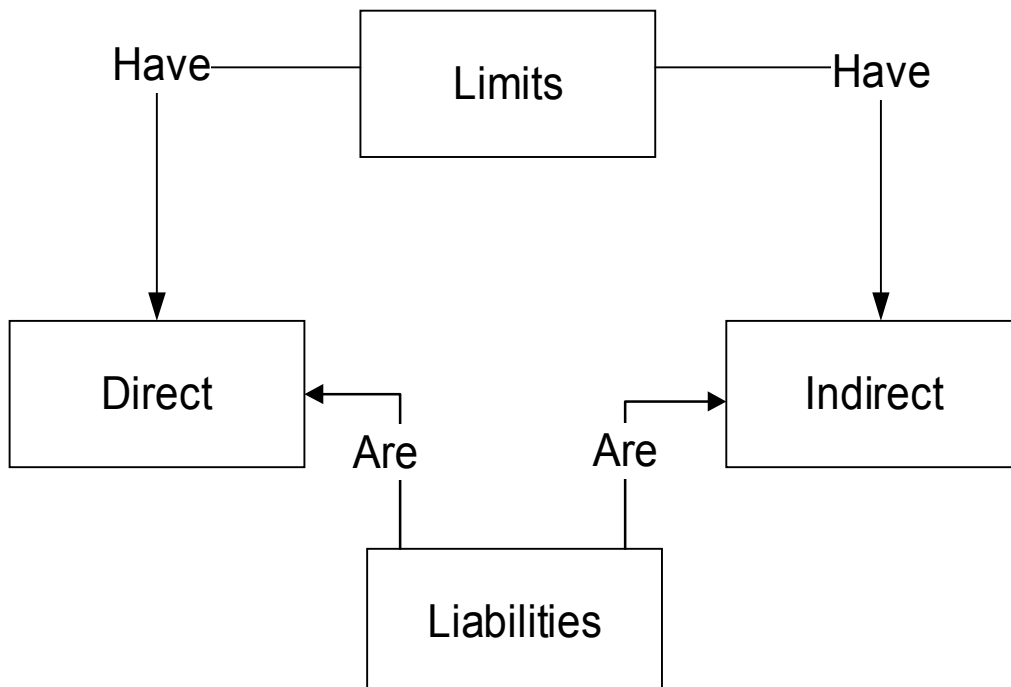
Rights are shown in Figure 30 (below). A cloud service user will have general rights of usage over a service, these are general rights to use and enjoy the service. A specialisation of general rights are proprietary rights that are enforced by property and contract law, they relate to total ownership of a cloud service and associated ownership aspects such as data ownership. Warranties are guarantees if the service malfunctions or causes damage.





**Figure 30 - Rights Terminology and Relations**

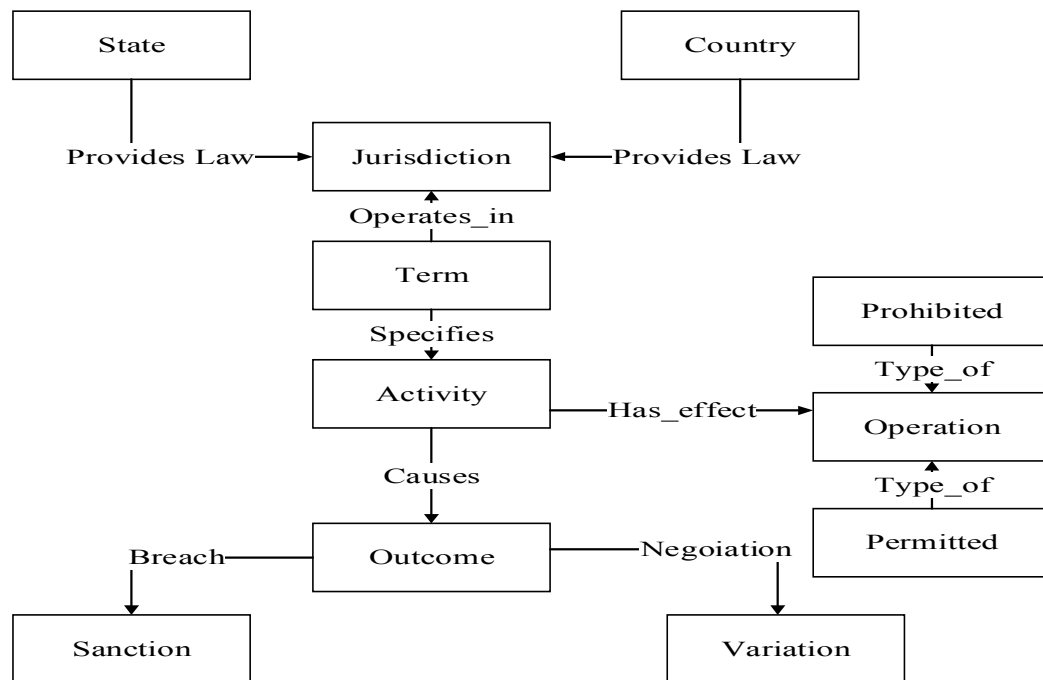
Liabilities are shown in Figure 31 (below). The liabilities of cloud service provider can be direct or indirect and limits are placed on the liabilities.



**Figure 31 - Liabilities Terminology and Relations**

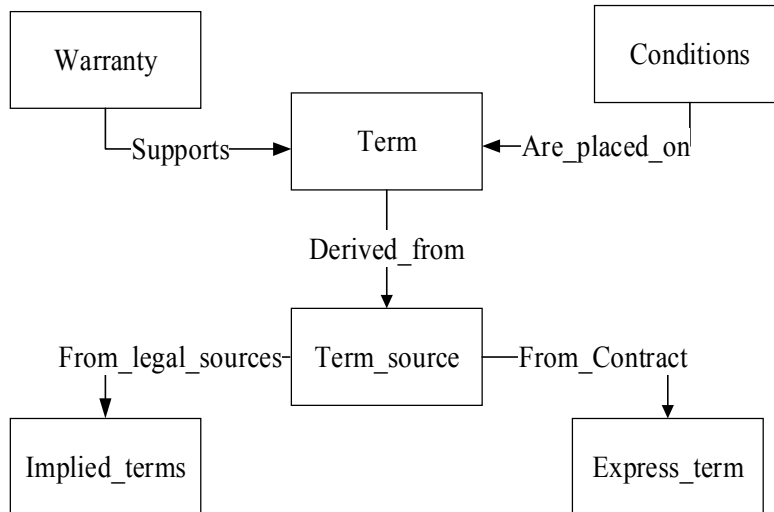
## 5.6 Acceptable Use Policy

Figure 32 (below) represents a high level TBox for Acceptable Use Policy (AUP) developed. Terms in the AUP are the central concept and are developed in accordance with state and a country's law. The terms of the AUP define activities which are prohibitive or permitted. When a customer performs an activity there is an outcome which may result in sanction or variation of the terms of the AUP.



**Figure 32 - Acceptable Use Policy TBox Overview**

The terms of the AUP are associated with a number of legal concepts. The terms may be expressly stated in the AUP or be implied through custom or dealing between the parties. Conditions and warranties may be attached to terms. It is important to specify these concepts, assertions and relationships in description logics as some ambiguity can be removed from the specification of the AUP. Figure 33 (below) shows the terminology surrounding terms.



**Figure 33 - Terminology surrounding AUP Terms**

The assertions (ABox) were developed for the terminology. A number of examples are shown in Figure 34 below.

Term(Anti\_Spam) “Don’t use service to send spam”  
 Activity(Sending\_Spam)  
 Specifies(Term:Anti\_Spam, Activity:Sending\_Spam)  
 Outcome(Service\_email\_overloaded)  
 Causes(Activity: Sending\_Spam, Outcome: Service\_email\_overloaded)  
 Sanction(Suspension\_of\_email)  
 Breach(Outcome:Service\_email\_overloaded,  
 Sanction:Suspension\_of\_email)

**Figure 34 - Acceptable Use Policy ABox**

The RBox provides a number of role-inclusions as shown in Figure 35 below.

Prohibited is a Type\_of Operation  
Permitted is a Type\_of Operation  
A number of role equivalences are defined:  
Prohibition Is\_equivalent\_to Specifies(Term, Activity) AND Has\_effect  
(Activity , Prohibited)  
  
Not\_Prohibited Is\_equivalent\_to Specifies(Term, Activity) AND Has\_effect  
(Activity , Permitted)  
  
Implied\_term Is\_equivalent\_to Derived\_from(Term, Term\_source) AND  
From\_legal\_source(Term\_source, Implied\_term

**Figure 35 - Acceptable Use Policy RBox**

The legal aspects of AUP and SLA are closely related and thus it may be possible to develop a common legal semantic description between the two CSA artefacts.

## **5.7 CSA Case Study**

### **5.7.1 Purpose**

A case study that demonstrates theoretical semantic framework for CSA will now be described.

### **5.7.2 System under Study**

Two CSA were examined for Oracle cloud services [167] and IBM [168] cloud services. The CSA were analysed to extract CA, SLA and AUP elements, the agreements were then encoded in terms of CSA terminology concepts and relationships.

### 5.7.3 Application of the Proposed Method

The terminology and relationships (TBox and RBox) developed for CSA were used to assertions (ABox).

### 5.7.4 Results

#### Oracle

**Requirements and Usage** will be mapped to a payment structure. Only metered services are shown for reasons of brevity.

The Oracle payment structure ABox is shown in Figure 36 below.

```
Payment_Structure(MeteredServicesGeneralPurposeStandardCPU,  
Cost($600, Month), Cost($1.008,Hour))  
Payment_Structure(MeteredServicesGeneralPurposeEnterpriseCPU,  
Cost($3000, Month), Cost($5.04, Hour))  
Payment_Structure(MeteredServicesGeneralPurposeHPCPU,  
Cost($4000,Month), Hour($6.72, Hour))  
Payment_Structure(MeteredServicesGeneralPurposeEPCPU, Cost($5000,  
Month), Cost($8.401,Hour))  
Payment_Structure(MeteredServicesHighMemoryStandardCPU(Cost($700,  
Month), Cost($1.176,Hour))  
Payment_Structure(MeteredServicesHighMemoryEnterpriseCPU(Cost($310  
0, Month),Cost( $5.208, Hour))  
Payment_Structure MeteredServicesHighMemoryHPCPU(Cost($4100,  
Month, Cost($6.888, Hour))  
Payment_Structure(MeteredServicesHighMemoryEPCPU(Cost($5100,  
Month), Cost($8.569,Hour))
```

**Figure 36 - Oracle Payment Structure ABox**

The Oracle Temporary Suspension ABox is shown in Figure 37 below.

```

Rule(Suspend_Account,
Customer_Action(significant threat to the functionality, security, integrity, or
availability of the Services or any content, data or applications),
Customer_Action(Illegal Act),
Customer_Action(violation of the Acceptable Use Policy),
Temporary_Suspension(deny_all_access_until_resolved))

```

**Figure 37 - Oracle Temporary Suspension ABox**

The Oracle Terms and Conditions ABox is shown in Figure 38 below.

```

Contract(Contract_for_cloud_database_services,
Term(Fees Due 30 days),
Term(Immediate payment overuse),
Term(receive multiple invoices),
Condition(Not(Cause_damage)),
Condition(Not(Send_Spam)),
Condition(Not(Perform_Benchmarking)),
Condition(Not(Disclose_Network_Ports)))

```

**Figure 38 - Oracle Terms and Conditions Abox**

The Oracle Security and Risk ABox is shown in Figure 39 below.

```

HasModel(Public_Cloud,Control(User, Risk(viruses)))
HasModel(Public_Cloud,Control(Control(User, Risk(Trojan Horses)))
HasModel(Public_Cloud,Control(Control(User, Risk(Worm)))

```

**Figure 39 - Oracle Security and Risk ABox**

The Oracle Acceptable Use Policy ABox is shown in Figure 40 below.

```

Sanction(HasJurisdiction(Any_Country(Cause_damage)), Suspend)
Sanction(HasJurisdiction(Any_Country(Send_Spam)), Suspend)
Sanction(HasJurisdiction(Any_Country(Perform_Benchmarking)), Suspend)
Sanction(HasJurisdiction(Any_Country(Disclose_Network_Ports)), Suspend)

```

**Figure 40 - Oracle Acceptable Use Policy ABox**

## **IBM**

The IBM Requirements and Usage are mapped to a payment structure are shown in Figure 41 below.

```
Payment_Structure(Standard_Capacity, CPU(96, Cores), RAM(384, GB),  
Storage(3.6), TB), Neworking(1, Gbps), Cost($7100, Month))  
Payment_Structure(Enterprise_Capacity, CPU(192, Cores), RAM(384, GB),  
Storage(3.6), TB), Neworking(1, Gbps), Cost($7100, Month))
```

**Figure 41 - IBM Payment Structure ABox**

The IBM Temporary Suspension ABox is shown in Figure 42 below.

```
Rule(Suspend_Account,  
Customer_Action(unlawful),  
Customer_Action(obscene),  
Custimer_Action(offensive),  
Custimer_Action(fraudulent),  
Temporary_Suspension(deny_all_access_until_resolved))
```

**Figure 42 - IBM Temporary Suspension ABox**

The IBM Terms and Conditions ABox is shown in Figure 43 below.

```
Contract(Contract_for_cloud_database_compute,  
Term(Fees Due 3 after invoice),  
Term(Late payment fees),  
Term(May require payment in advance),  
Condition(Not(send viruses)),  
Condition(Not(unsolicited messages)),  
Condition(Not(send harmful code)),  
Condition(Not(resell cloud access)))  
Condition(Not(combine cloud services)))
```

**Figure 43 - IBM Terms and Conditions ABox**

The IBM Security and Risk ABox is shown in Figure 44 below.

```
HasModel(Private_Cloud,Control(User, Risk(viruses)))  
HasModel(Private_Cloud,Control(Control(User, Risk(unsolicited messages)))  
HasModel(Private_Cloud,Control(Control(User, Risk(harmful code)))  
Sanction(HasJurisdiction(Any_Country(Cause_damage), Suspend)
```

**Figure 44 - IBM Security and Risk ABox**

The IBM Acceptable Use Policy ABox is shown in Figure 45 below.

```
Sanction(HasJurisdiction(Any_Country(send viruses)), Suspend)  
Sanction(HasJurisdiction(Any_Country(unsolicited messages)), Suspend)  
Sanction(HasJurisdiction(Any_Country(send harmful code)), Suspend)
```

**Figure 45 - IBM Acceptable Use Policy ABox**

### **5.7.5 Evaluation**

The two CSA can be modelled using the proposed semantic framework by examining the CSA documents supplied by the cloud service providers and other publically available information they provide. The case study does not deal with all the information generated by providers, for the sake of brevity.

The Oracle payment structure is vague compared to IBM payment structure and, will require further development, so that it can be mapped to requirements.

The temporary suspension assertions are quite similar for both Oracle and IBM and breach of conditions result in all services being suspended.

Terms mainly deal with payment rules. Conditions are prohibitive and can be mapped to security risks and AUP sanctions.

Security risks are concerned with sending spam and viruses and are strongly mapped to the contract conditions and the AUP. The term spam in



the Oracle CSA can be mapped to ‘Unsolicited Messages’ in the IBM CSA, as they are semantically equivalent.

The AUP takes the contract conditions and security risks and describes actions (for example sanctions) in a given jurisdiction.

In conclusion, the semantic framework can be used to model real world CSA. This allows CSA to be compared and mapped. A clear pattern emerges from the CSA. Once a payment structure is resolved to CPU, RAM, Storage, Networking and Cost, then requirements can be mapped to usage specifications. The drivers for security, risk and AUP are prohibitive conditions. Common terms such as ‘Spam’ and ‘Unsolicited Messages’ are semantically equivalent terminology.

A clear workflow and semantic representation for analysing CSA is highlighted from the case study, this provides new theoretical knowledge in the area of cloud service description.

## **5.8 Comparison to other Cloud Service Agreement Frameworks**

Brandic et al. [3] VieSLAF present VieSLAF a SLA mapping service based on template mapping. The approach utilises templates developed as XML descriptions. The VieSLAF service processes SLA documents formatted using XML template specifications. There is an attempt to map the templates using style sheet transformations (XSLT), which only provides simplistic syntactical mapping. Semantic mapping using technologies such as SPARQL [169] is more powerful than the syntactical mapping offered by XSLT which can take place across terminology, relationships and assertions. Powerful semantic description mapping tools and fuzzy matching approaches have been developed by a number of researchers [170].

Oldham et al. [171] describe web-service agreement partner searches using semantic matching. The researches highlight the deficiencies in the Web-Service Description Language (WSDL), such as the lack of modelling for Quality of Service (QOS) and pricing of services. The approach models Web-Service Agreements (WSA) as a number of Service Level Objectives (SLO). The WSA are matched by algorithms and rules. This approach could be extended for use with cloud computing.

Grozev and Buyya's work on SLA describes a combination of brokerage approaches that are required to provide a match between a user's demands and service provider's offerings. The researchers identified a number of challenges in current cloud architectures that affect Quality of Service (QoS). Garg et al. [172] describe a number of possible criteria for QoS, such as timeliness, availability and reliability of cloud services. The provision of cloud services could be configured with a high base provision of reserved resources, which allows the service provider to deliver to customers with a high QoS regardless of demand or with a QoS that exceeds their SLA. The problems with this approach are the cost and wastage of perishable cloud resources.

To overcome the issues highlighted by Grozev and Buyya [32] an 'intercloud' architecture is proposed. The architecture provides a cloud model that guarantees service quality and allows the reassignment of resources and the coordination of consumers' requirements to meet defined SLAs.

## **5.9 Discussion and Conclusions**

This chapter has outlined the contents of Cloud Service Agreements (CSA) which formalise the relationship between cloud service users and providers. The three artifacts of CSA, Customer Agreements (CA) Service Level Agreements (SLA) and Acceptable Use Policy (AUP) were modelled as semantic framework based on description logics. The benefits from this

research are a unique complete formal specification of all aspects of CSA. The semantic description developed has been utilised by other researchers and could be used by practitioners to analyse individual CSA artifacts or develop software applications to automatically analyse CSA.

A major benefit of using semantic description is ability to merge and cross reference common concepts and relationships of the three artefacts of the CSA. The semantic description can also be merged and linked to other semantic description related to CSA, for example, legal and contract semantic description.

The semantic framework approach provides the opportunity to create a complete semantic description for cloud service description using a collaborative semantic description development methods, using toolsets such as WebProtégé

Using Description Logic based semantic description provides a clear, precise and unambiguous specification of CSA which can be reasoned across to create new knowledge. This allows new CSA to be analysed quickly and to provide cloud service consumers with analysis framework.

Semantic description may be difficult for naïve users to understand and develop, however, the semantic description can be used in a software solution with a user interface that allows the user to enter and search (ABox) assertions based on the framework's terminology (TBox and RBox).

## **Chapter 6      Requirements for Semantic Description of Cloud Computing Services**

### **6.1 Chapter Overview**

Requirements follow Cloud Service Agreements. A specialised semantic approach of Problem Solving Ontology is used to define high level requirements for cloud service requirements.

High level requirements link to low level requirements, which in turn map to cloud service resources.

A case study is presented to demonstrate the approach.

### **6.2 Introduction**

This chapter describes the semantic framework for cloud computing service requirements and discusses an ontology design based on the framework.

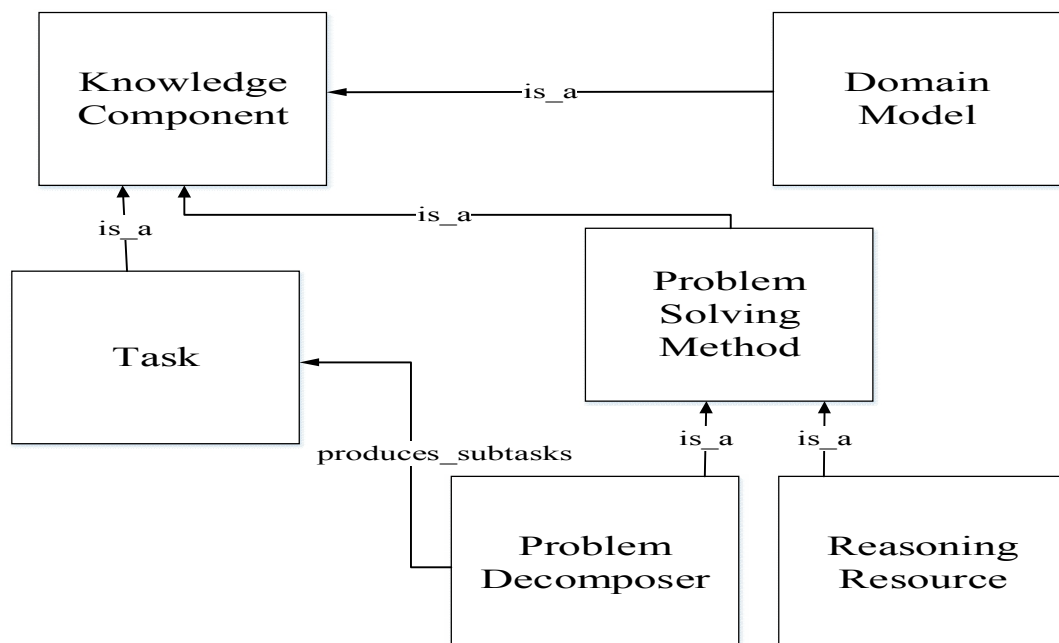
### **6.3 Requirements Semantic Description Design**

A detailed requirement can be mapped to a number of 'knowledge components' for implementation within semantic description modelling tools. The knowledge component provides a base selection of properties such as description and requirement pragmatics. Elements of the semantic description then inherit properties from the knowledge component. Specialist Problem Solving Methods (PSM) such as problem decomposers (PSMs that can split a task into subtasks) can be developed for specific purposes. Requirements engineers can develop their own specialist tasks, PSMs and domain models for a specific requirements problem using powerful mechanisms such as inheritance and set operations. The usage of tasks,

PSMs and domain models will lead to greater reuse as a generic method PSM.

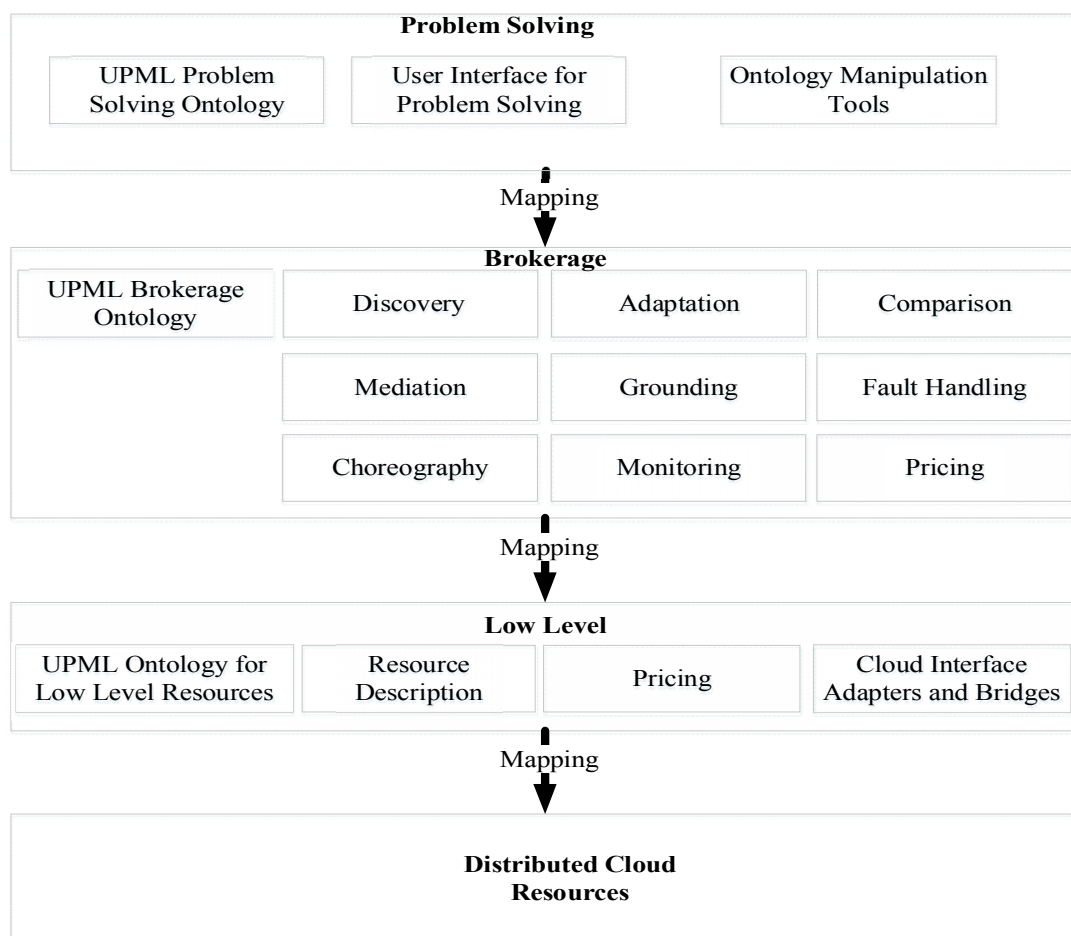
The Figure 46 below describes a model into which requirements can be tailored. This machine readable model is used directly in cloud computing environments.

The semantic description provides a checklist of 'what' requirements are needed and is specified in terms of tasks, PSMs and domain models. The model provides representation for elements of requirements. Requirements are expressed in terms of semantics and, concepts such as tasks can be expressed in terms of rich semantics, as can relationships between tasks, PSMs and Domain Models. This allows the requirements engineers' greater expressive power, and the ability to carry out fuzzy searches, to map new knowledge and requirements via the reasoning tools seen in semantic description toolsets.



**Figure 46 - Requirements Semantic Description Implementation**

The architecture of the semantic description is described in the figure below. The highest layer deals with problem-solving for cloud computing. Users will have tasks which use the PSMs and domain semantic description. The brokerage layer defines elements in terms of semantic description, tasks will be executed at a strategic level across the cloud environment dealing with issues such as cost and quality of service. The low level layer deals with operational requirements mapped to cloud service providers' interface descriptions.



**Figure 47 - The Hierarchical Structure of the Semantic Description**

The details of each element of the requirements semantic description from Figure 47 above will now be described.

## **6.4 Problem Solving**

The problem-solving layer relates to the high level requirements of users expressed as semantic description and, they describe 'what' is required; which may be full requirements or partial requirements expressed as problem solving semantic description fragments. The requirements must be matched to low level requirements through the brokerage process via direct or fuzzy matching.

The separation of requirements into tasks, PSMs and domain knowledge and representation as UPML provides ease of mapping to low level requirements through the brokerage process.

## **6.5 High Level Requirements - Brokerage**

The brokerage of requirements map high level requirements to lower level requirements. This is carried out by semantic searching, fuzzy matching and negotiated processes. Discovery can be driven by the Quality of Service (QOS) requirements. These requirements are carried forward from user requirements expressed in the high level layers of the requirements semantic description. Monitoring can use the QOS requirements to define the requirements for service failure. Pricing requirements can also be related to QOS.

### **6.5.1 Discovery**

Discovery requirements describe what and how cloud services are found across a set of cloud resources. Adaptation requirements relate to how defined requirements can be adapted to meet new user requirements. The adaptation process is made easier by the use of semantic description as sets of requirements that that can be adapted by recombination through semantic relationships provided in the UPML semantic description. Closely

related to adaptation, composition requirements describe how sets of cloud resources are combined to meet high level user requirements.

### **6.5.2 Mediation**

Mediation requirements define how high level problem-solving requirements will be translated into low level requirements by a process of iteration. Many of the mediation requirements are concerned with QOS, Rimal et al [173] describe the need for quality of service requirements in cloud computing. Quality of service provides a guarantee of the availability and performance of tasks inside a cloud computing service. Requirements are supported by service elements such as security, reliability and dependability. Stakeholder groups will place value on service elements, for example low latency short burst resources will be required by some users, whereas other users will require long running resource pools. Grounding requirements link the execution of the requirements with how the requirement is to be executed at a low level. Fault handling requirements provide actions that are necessary when errors occur at lower levels in cloud description framework.

### **6.5.3 Choreography**

Choreography requirements provide the approach required for coordinating higher level requirements so they are performed correctly at a low level. Monitoring requirements specify the information required as tasks are executed and choreographed. Pricing requirements at the brokerage level deal with pricing estimation for a given high level tasks and aggregate pricing for packages of low level tasks.

The requirements semantic description can draw upon many leading research concepts seen in the literature to represent concepts in the requirements semantic description as a complete semantic description or semantic description fragments. Robinson [174] describes service monitoring, which is a brokerage component within the requirements



semantic description. The high level requirements for monitoring can be represented as a PSM, shown in the list below:

- Define the design-time model
- Define goals and requirements
- Define obstacles and monitors
- Define the run-time model
- Monitor the running program

It should be noted that this PSM can be used by a number of tasks as the PSM can act on a number of problem domains. The requirements are represented in the brokerage layer. A primary goal can be decomposed by a specialist PSM called a 'problem decomposer'. Tasks such as 'monitor' will have inputs, outputs, competencies and formal definitions seen in figure above. Lower level representation can also be represented as semantic description.

The discovery and monitoring processes can use a similar service discovery and monitoring approach in cloud computing. High level requirements are used to drive the service discovery of web-services. Users can then select cloud services that match their QOS requirements.

#### **6.5.4 Adaptation**

Service adaptation can be seen in the requirements semantic description framework. Higher level requirements goals and services categories can be represented as domain models; these are measured using a 'measure' PSM. In the brokerage layer service definitions are domain models used by a monitor definition PSM. The lower service domain models are monitored by PSM.

Semantic representation could provide many of the features required by researchers, such as fuzzy searching and the matching and representation of partial requirements using ontological fragments.

### **6.5.5 Grounding**

Grounding relates to the relationships between high level and low level requirements and between low level requirements and the cloud service providers' service specification.

Semantic representation is used to map high level requirements via intermediate concepts and relationships, for example a high level problem solving method will be mapped via brokerage to a software program, along with notational CPU (CPU demands from a standard reference, for example an Intel I3 processor which can be adjusted for other notional CPU types) and storage requirements.

Low level requirements are mapped to cloud service specifications via brokerage with concepts such as notional CPU mapped to actual CPU selections.

### **6.5.6 Monitoring**

High level monitoring requirements will comprise the elements shown in the list below:

- Monitoring descriptions of tasks that are mapped to low level notional software descriptions, including CPU, memory and storage requirements.
- Key performance indicators for software run times, CPU and memory limits and network latency
- Descriptions of possible breaches of Acceptable Use Policies (AUP)

### **6.5.7 Comparison**

Concepts and relationships for how tasks, problem solving methods and knowledge domains are semantically compared.

### **6.5.8 Fault Handling**

High level descriptions of actions for fault handling shown in the list below:

- A task mapped to resources fails or doesn't start
- Resources are not available to run a task
- Failure of choreography between tasks

### **6.5.9 Pricing (High Level)**

A high level pricing description will be generated independent of the low level service provision. This price will be based on mapping to low level concepts such as notional CPU costs, memory costs and storage requirements. Feedback from low level service descriptions mapped to actual services will update high level requirements.

## **6.6 Low Level Requirements**

Resource description requirements of each low level cloud resource are required so that they can be brokered. Examples of a resource description could be maximum CPU capacity, storage capacity, response time and spare CPU capacity.

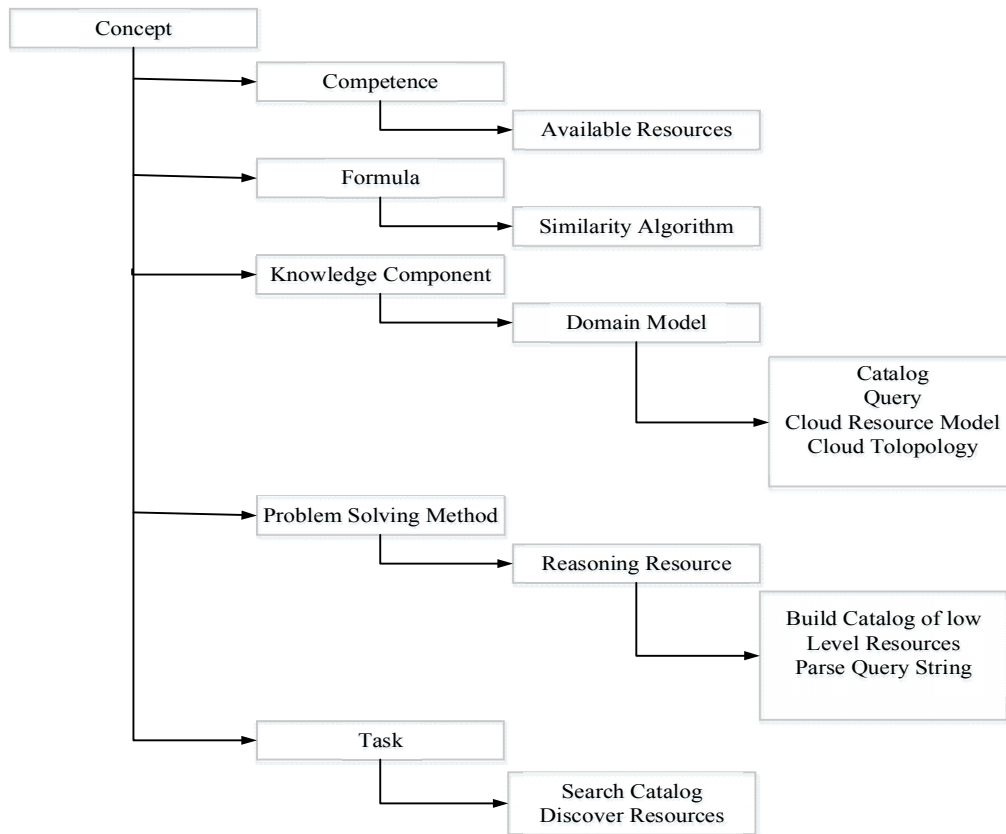
Sun et al [175] point out that cloud computing has seen vendors offering a number of cloud computing platforms. Semantic description can be used to describe vendors' offerings and can be used to abstract models from the integration of disparate offerings. Pricing requirements at a low level deal with areas such as the cost of CPU capacity and storage capacity. The

requirements of cloud adaptors and bridges provide information for brokerage requirements.

## **6.7 Specification**

The three levels of the requirements semantic description (problem-solving, brokerage and low level) are all described in terms of UPML. This allows mapping, stepwise refinement, interaction and reasoning to be carried out between the layers. The usage and processing of semantic description fragments has been described by a number of researchers [176], [177] and [178].

The high level problem-solving requirements are specific to each individual requirements domain or process. They are still defined and structured in a UPML and the example of high level problem-solving is shown in the case study, shown in Figure 48 (below). The requirements semantic description concentrates on the brokerage and low level aspects of Requirements Engineering (RE). Brokerage fragments will take the problem-solving requirements and consider the requirements for their fulfilment. An example of a brokerage fragment will be given for discovery. Discovery is the process of finding resources for the fulfilment of a high level requirement. In the figure below, the RE semantic description fragment for discovery is shown. The UPML semantic description provides the framework for semantic description fragments, which in turn guide the subsequent RE process. The discovery process is driven by two tasks, the discover resources search the cloud resource model and the cloud technology to build a catalog of resources and will search the catalog with a query string allowing the resource discovery.



**Figure 48 - Discovery Element Semantic Description Fragment**

The Table 7 below shows how properties can be defined for the “Discover Resources” task.

Input/Output Class	Cardinality	Type
Input	Exactly 1	Cloud Resource Model
Input	Exactly 1	Cloud Topology
Output	Exactly 1	Catalog

**Table 7 - Discover Resources Task**

Now the requirements has been described in detail a case study will demonstrate how the requirements can be defined for each requirements semantic description element.

## **6.8 Requirements Case Study**

### **6.8.1 Purpose**

The case study shows how RE can utilise a UPML based semantic description using the concepts described in the requirements semantic description.

### **6.8.2 System under Study**

The case study describes the requirements for a document similarity framework, which allows documents such as academic texts to be compared to the research they reference. Manning et al [163] outline an approach for document similarity, shown in the list below.

- Collect the documents to be indexed
- Tokenise the text
- Carry out linguistic pre-processing of tokens
- Index the documents that each term occurs in
- Use similarity measures based on mathematical measures, such as Cosine Similarity [163]
- Report or carry out further processing such as clustering

The case study is particularly suited to cloud computing service description as large amounts of parallel processing are required to process documents. In single processor machines finding and comparing thousands of documents can take several hours. There is also scope to expand the application to recursively find referenced documents from the documents referenced from a study text.

### 6.8.3 Application of the Proposed Method

Using the “The Hierarchical Structure of the Semantic Description” shown in Figure 42 (above) each element was identified for a series of high level tasks, using the UPML representation (Appendix B). A mapping strategy was developed for high and low level requirements.

### 6.8.4 High Level Problem-Solving Requirements Ontology

The high level tasks required for the case study are described in the Table 8 (below). The requirements describe a workflow of tasks which need to be executed to carry out document similarity for a document from a student course.

Task Requirements	Description
Find academic references in course Documents	Parsing to find document references.
Create structured references and import into a reference management system	Format references so they are machine readable.
Find academic research for references	Find references automatically using cloud-services
Extract plain text from the PDF files, break into pages and tokenise text	Use off-the-shelf cloud software libraries
Pre-process tasks and indexation	Use off-the-shelf software libraries
Create similarity measures and match documents	Suited to Cloud computing Burst of processor bound tasks
Reporting	Report document similarity

**Table 8 - Case Study: High Level Requirements**

These task requirements are then converted into UPML semantic description. The requirements are split into tasks, PSM and Domain Models.

#### **6.8.5 Brokerage**

The brokerage semantic description matches the high level requirements to the low level requirements semantic description. Each aspect of the semantic description will now be discussed.

#### **6.8.6 Discovery**

Discovery can be seen as the high level requirements which use a 'find low level requirement for a high level requirement' Problem Solving Methods (PSM). The high level requirement task 'evaluate\_corpus' requires the low level formulas such as 'Ratio Distance' and will discover 'match\_research\_to\_study\_text\_research'.

#### **6.8.7 Adaptation**

Adaptation is the process of adapting low level requirements to meet a new or existing high level requirement. Composition defines the ordering of requirements tasks to complete the goal of producing document similarity for a corpus of documents. The tasks in the case study are self-organising as output from one low level resource feeds the input of another low level resource.

#### **6.8.8 Mediation**

Mediation is driven by the high level requirements specification to find the most appropriate low level resource by stepwise refinement.

An off the shelf choreography model was used. Yazir et al [179] describe the PROMETHEE methodology for choreography across multiple cloud resources where a number of physical machines (PM) are allocated including Virtual



Machines (VM) across a set of cloud resources. Monitoring requirements concern the information required being used to review the progress of the low level execution of tasks.

An existing pricing model was used in this case study. Henzinger et al. [180] discuss the Flexprice model for pricing across multiple cloud resources. In a commercial implementation of a document similarity framework high level task requirements will be priced across a number of cloud providers and the most cost effective solution will be selected.

### **6.8.9 Grounding**

Grounding is a simple mapping of high level tasks requirements to individual software modules. In the document similarity example 'collect documents to be indexed' will be mapped to a number of groundings, shown in the list below:

- A problem solving method 'indexer' with
- A notional CPU requirement
- Memory requirement
- Storage requirement

### **6.8.10 Monitoring**

The monitoring requirements are shown in the list below:

- If any of the four tasks fail (Collect, Tokenise, pre-process and Index) the user should be notified
- If resources are not available to run the tasks the user should be notified
- Failure of choreography between the four tasks will be notified to user

### **6.8.11 Comparison**

Mappings will be made from the high level resource requirement to low level resource descriptions.

### **6.8.12 Fault Handling**

Fault handling requirements deal with actions that occur in low level programming language modules, virtual machines and physical machines.

### **6.8.13 Pricing (High Level)**

Pricing will be defined for each of the tasks. This will be mapped to detailed pricing information in the low level requirements. Tasks are related to generic problem solving methods that have been previously instrumented to provide notional CPU, memory and storage requirements.

### **6.8.14 Low Level Requirements Semantic Description**

A number of formulas are required to calculate document similarity. UPML allows individual software components to be described. Tasks describe the operations required to meet requirements high level requirements. UPML can describe both high and low level requirements in a structured way.

### **6.8.15 Mapping**

The mapping process uses semantic search technologies such as SPARQL [133] and fuzzy matching to match high level brokerage requirements to low level requirements which describe distributed cloud services at an abstract level.

The mapping process for discovery will use a special PSM, a decomposer (Shown in Appendix B) to decompose the 'evaluate\_corpus' requirement into a number of sub requirements. The first sub-requirement is 'Calculate

Measure' which will match PSM for calculating measures which have the property measure, which is contained in a number low level PSM formulas such as Cosine Similarity and Jacquard Similarity represented as UPML ontology. Property measures are connected to 'match\_reasearch\_to\_study\_text\_research' so the high level requirement 'evaluate\_corpus' will be mapped to the low level requirement 'match\_research\_to\_study\_text\_research'. The Cosine Similarity and Jacquard Similarly [163] will be mapped to code modules in distributed cloud resources.

Adaptation of services take high level requirements and map these requirements to low level adapters, bridges and refiners shown in Appendix B. These low level requirements and bridges control the adaption of cloud services.

Comparison requirements map onto groups of low level resource descriptions, for example similarity measures provide a semantically equivalent service and therefore can be compared as being semantically equivalent at a high level.

Mediation and Grounding are iterative processes. A high level requirement such as 'Calculate Measure' will be mapped to resource descriptions which describe cloud services for a number of measures.

High level Fault Handling and Monitoring requirements map directly onto to low level resource descriptions defined as UPML ontology.

Choreography is purely a high level requirement that defines the order high level requirements are executed in.

High level pricing requirements will aggregate low level pricing requirements. A high level requirement such as 'Find academic references in course Documents' will map to costs for a number of low level search resources.

### **6.8.16 Results**

It was possible to match tasks from the case to study to the elements seen in The Hierarchical Structure of the Semantic Description” shown in Figure 47 (above). High level requirements can be mapped to low level requirements, with low level requirements mapping to physical cloud resources.

### **6.8.17 Evaluation**

The case study has demonstrated the requirements semantic description built on UPML. The three layers of the requirements semantic description provide guidance for the definition of a document similarity framework for study texts and the research referenced from the study text. High level requirements, brokerage and low level requirements are expressed as textual requirements and, then as a UPML semantic description. Semantic description mapping and reasoning tools can be used to match each layer of the model, so that high level requirements can be executed by appropriate resources in the cloud. The use of semantic description leads to a greater reuse of requirements and the generation of new requirements by reasoning.

## **6.9 Comparison to other Cloud Computing Requirements Frameworks**

Wind and Schrödl [44] propose traditional software requirements engineering tools such as V-Model, Rational Unified Process, Volere and Extreme Programming for Requirements Engineering (RE) in cloud computing. Although these approaches have some value in capturing cloud service requirements there are many issues. Firstly, customer self-service is a major requirement for cloud computing services. A requirements approach must be able to showing the similarities and differences between semantically similar services with are quite different at a syntactical level.

The tools proposed by Wind and Schrödl do not have the level of expression to allow such comparison. Secondly, many of the requirements are derived from agreements between cloud service providers and users. The traditional requirements frameworks do not consider such agreements. Thirdly, pricing is another area of requirements that software engineering requirements frameworks do not contemplate.

Rimal et al. [173] is presents typical research into to requirements for cloud computing. There are two major deficiencies in the approaches seen. Firstly, although the researchers present a number of relevant topics for requirements such as security, Service Level Agreements (SLA), privacy and data governance. There is no mechanism for encoding requirements into a framework as with the proposed framework promulgated in this thesis.

Secondly, Rimal's framework concentrates too much on low level requirements such as virtual machine, infrastructure and data storage with no mapping to high level requirements.

## **6.10 Discussion and Conclusions**

Requirements engineering semantic description provides a three layer framework for RE in the cloud computing environment built on UPML.

The semantic description can be checked for correctness and reasoning and can map new knowledge from the semantic description that can be relayed to users. Requirements can be inserted into the semantic description and used at a later date. Requirements can be found using semantic or fuzzy searching as well as syntactical searching.

The requirements semantic description environment can be used to develop meta-services. These meta-services support two key features that are new to cloud computing self-service and on-demand provision. The high level

and brokerage requirements seen in the requirements semantic description allow customers to access on-demand self-service via meta-services..

The reuse of requirements is a key advantage of using a UPML based semantic description. A PSM can be used in many knowledge domains and knowledge domains can be re-used for new requirements. Problem-solving semantic description frameworks are seen as useful for cloud computing as it can be seen as a problem-solving paradigm, as opposed to an extension of SaaS or virtualisation of existing applications.

The requirements engineering problem is broken down into three sets of concepts: tasks which describe the work that is to be done, problem-solving methods which describe the solutions to problems, and a problem domain which describes concepts for a given requirements scenario. The semantic requirements description builds on a UPML structured semantic description approach across the three distinct levels in cloud computing RE. Semantic description mapping is seen as a key tool for linking requirements at a number of levels in the requirements semantic description framework.

## **Chapter 7      Semantic Description of Cloud Service Pricing**

### **7.1 Chapter Overview**

Pricing of cloud services is becoming increasingly complex with a number of pricing strategies (pre-purchase, pay-as-you-go and spot markets), across a number providers and geographic locations.

To allow comparisons between cloud service choices a sophisticated description is required. A semantic description of cloud computing services allows users to select the most cost efficient service choice.

A case study is presented that demonstrates the approach.

### **7.2 Introduction**

A semantic pricing framework can be achieved by building on problem solving semantic description approaches and implementations described in the literature review and related work respectively.

An example of Hybrid Meta Heuristics (HMH) will be described to demonstrate the benefits of using a semantic description driven approach. Powerful rule specification using [144] and queries using SPARQL [169] allow software systems to be created around the ontology implemented from the semantic description framework.

The ability to use off the shelf Problem Solving Ontology (PSO) and ability to overlay semantic description provides the correct modelling abstractions, good reuse, the ability to merge terminology with other semantic description frameworks and the ability to generate a number of user world views.

Cloud pricing strategies are analysed to build a problem-solving semantic description for pricing. Bhargava and Sundaresan [181] describe utility provision of computer services with prices set by an open market with a number of commitment models. Organizations with predictable demand will be able to purchase resources in advance. A number of strategies are presented. Advanced Commitment (AC) customers pre-book resources and lose their deposit for non-use. No Commitment (NC) pay-as-you go customers. Part Commitment (PC) agreement to purchase resources which can be added to. Non-usage may be refunded. Amazon's commitment model is similar to the approach used in Bhargava and Sundaresan. An interesting commitment model recently introduced is described as a spot market, which provides an auction for cloud resources which that relate to the work of Martin et al. [182]. These models are driven by price, with discounts for advanced commitment and purchasing resources on spot-markets. The discounts obtained on purchasing cloud resources will be balanced against wasted resources.

Cloud pricing is formulated by deciding high level requirements and how the organization commits to implementing the requirements over time. High-level cloud computing requirements have been modelled as problem solving semantic description by researchers such as Bogg et al [49]. High-level user requirements are mapped to low level resource requirements.

The processing power of the machine instance is specified in terms of CPU power and memory size, vendors such as Amazon offer packaged machine configurations for a number of applications such as database, compute and storage. The processing speed and number of cores and memory size will increase the price of the instance selected. Utilization is an estimate of how much of the resource will be used in the selected period. A user would have difficulty with estimating the type of instance required. Data transfer and download expense are related to the setting-up data for processing and collecting results. The geographical location of a selected instance will affect



the price of the instance and the network latency of the data transfer, for example resources located in Europe may be more expensive than those located in the USA. Taxation and regulatory environments will also affect price.

An example of differentiation in price based on geographical location is shown in Table 9 (below). Pricing for a sample of simple Amazon EC2 LINUX instances shows the price differentiation between the USA, Europe and Japan [1]

Name	CPU	Memory	Virginia	Ireland	Tokyo
		(GB)	\$ US per Hour		
t2.small	1	2	0.026	0.028	0.040
t2.med	2	4	0.052	0.056	0.080
t2.large	2	8	0.104	0.112	0.160

**Table 9 - Pricing Differentials Based on Geographic Location**

Huang et al [183] examine hybrid approaches to pricing of cloud services. The researchers examine pricing of cloud services delivered by a mixture of service instances (reserved, spot and on-demand instances). Although the pricing of services delivered by a mixture of such services is more complex than the usage of single service instances, the benefit to both customers and vendors is increased. The situation is further complicated when services from several vendors are used to meet a set of user requirements.

### 7.3 Work flow design

The aim of the workflow is to use semantic description extended from the problem solving semantic description, to demonstrate how tasks can be priced using mappings to problem solving methods and domain ontologies, to provide high level pricing information.

Cloud service information requests provide the information processed by low level low cloud API's such as OCCl [184] to deliver initial cloud service information and also feedback information when cloud service requests are made.

Before a cloud service request is submitted for execution the pricing implications of the intended action are explained by reasoning against both high level requirements derived from high level task specifications and domain semantic description and low level information described as semantic description derived from low level APIs, such as OCCl.

High level requirements are broken into a series of tasks, which in turn can be broken into micro-tasks, the increased granularity of task specification will lead to better pricing decisions, as pricing can be spread across a number of number of cloud resources that provide the best prices for a given task linked to a requirement.

A task brings together problem solving methods for solving requirements described in tasks. Problem solving methods are standard algorithms for solving problems for given requirements, for example metaheuristics such as ant colony optimization algorithms. These algorithms can be combined and their characteristics in terms of time and resource usage are known and, therefore can be priced against a given payment structure. A domain semantic description provides information on a specific domain such as financial services calculations or genetic information processing, this will also contain an organization's preference for processing time, resource availability requirements and cost sensitivity.

Tasks are specified in terms of processing usage, memory usage, storage usage and transfer usage. Processing usage is the amount of CPU required to carry out the defined tasks. A number of academic sources such as Talbi [185] provide taxonomies and processor usage metrics of metaheuristics and their combination into hybrid- metaheuristics. Memory usage of tasks is

also strongly coupled to problem solving methods, again algorithm choice governs theoretical memory usage. Processor and memory usage will also be mediated by the domain semantic description as an organization trade higher processor and memory usage for speed of processing.

Debels et al. [186] provide CPU data for their hybrid scatter search/electromagnetism (EM) meta-heuristic for project scheduling the HMM model which is shown in Figure 49 (below).

```
Algorithm EM(maxiter, LSiter)
```

```
    iter :=1
```

```
    while iter<maxiter do
```

```
        local(LSiter)
```

```
        compute_forces
```

```
        apply_forces
```

```
        iter++
```

```
    endwhile
```

**Figure 49 - Electromagnetism (EM) Meta-Heuristic**

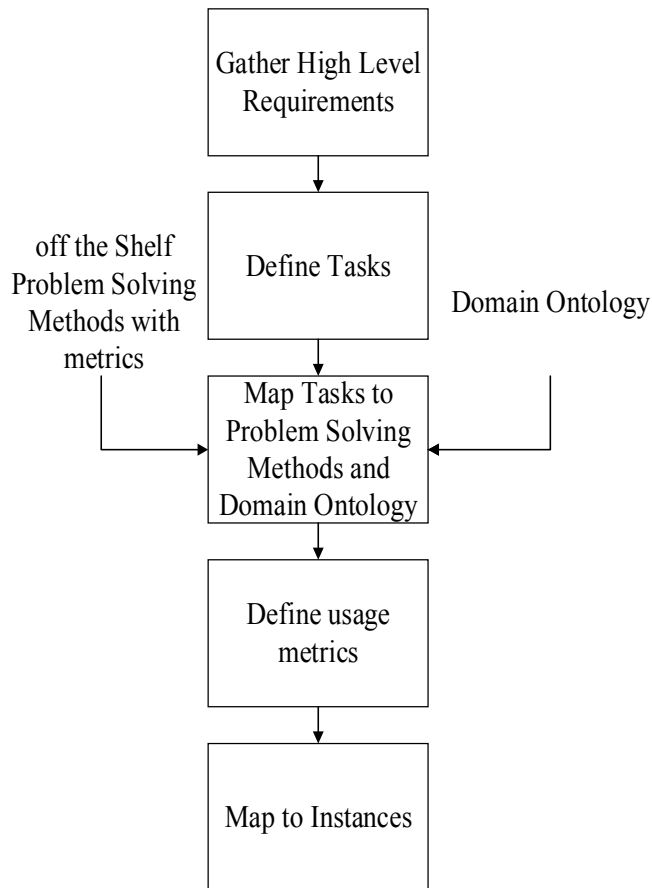
The CPU usage data is for a single PC running against a number of standard datasets (J30, J60, J90 and J210). Each dataset was tested against a number of schedules shown in the Table 10 below, which is instrumented against a number of cloud instances.

Problem Set	Schedules	J30	J60	J90	J120
Average CPU (Seconds)	1,000	0.02	0.06	0.14	0.21
	5,000	0.11	0.30	0.61	1.01
	50,000	1.10	3.02	6.08	10.18
	500,000	10.96	30.17	60.95	102.82
Maximum CPU (Seconds)	1,000	0.05	0.12	0.34	0.37
	5,000	0.17	0.48	1.01	1.72
	50,000	1.57	4.56	10.11	15.29
	500,000	14.60	46.78	100.36	155.04

**Table 10 - CPU Requirements for EM HMH**

Storage usage is driven by domain semantic description in terms of storage size and strategy. High volume manufacturing industries will require large amounts of streaming storage suited to NoSQL storage, a financial application may require smaller amounts of SQL database resources or simple structured storage resources. Simple storage may be suitable for genetic information processing where data is only stored as input for sophisticated processing. Transfer usage is the transmission of data from and to users. Transfer usage costs may be driven not only by transfer volumes, but also the need to use cloud resources in specific geographical locations for speed, security, legal and taxation reasons.

The pricing workflow for mapping high level requirements is shown in Figure 50 below.



**Figure 50 - Workflow for Pricing Cloud Computing Requirements**

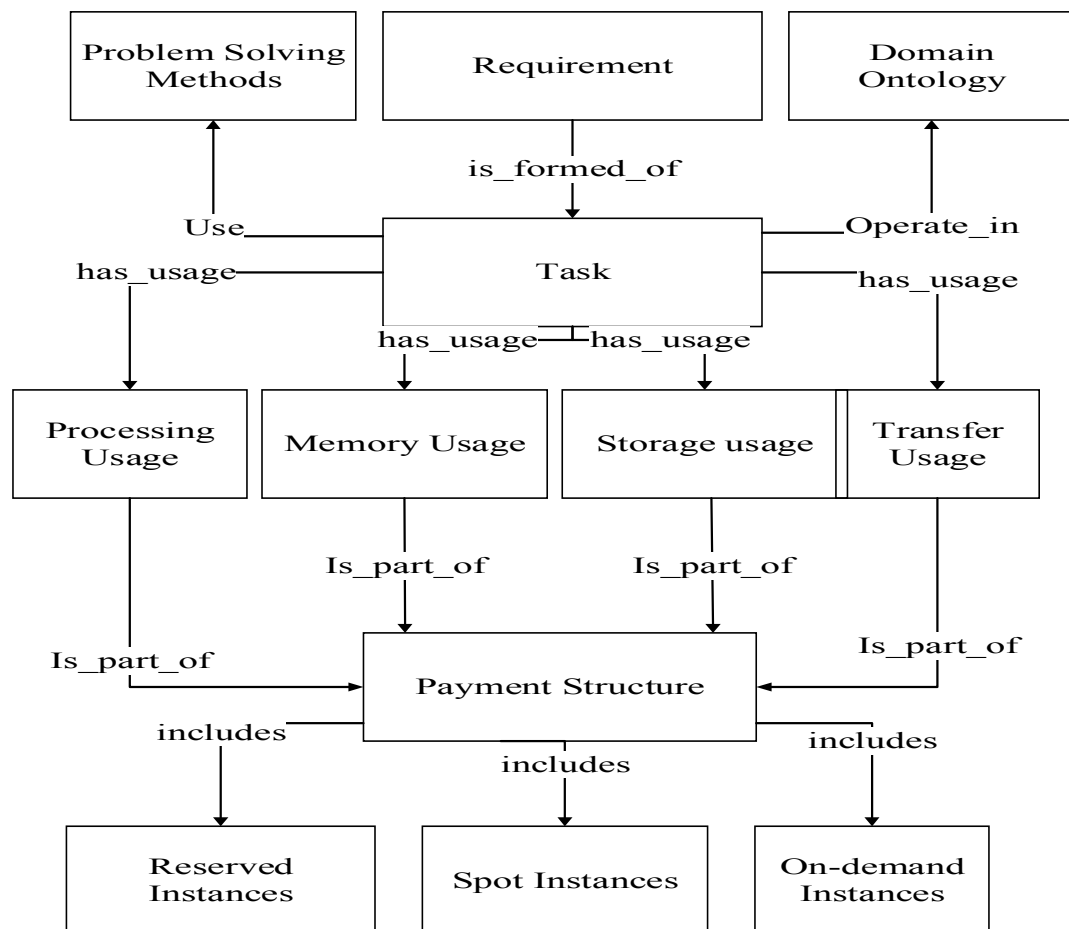
Theoretical payment structures are gathered from cloud service providers and are expressed in terms of generic semantic description. Processing usage, memory usage, storage usage and transfer usage will be mapped to the payment structures. Payment structures will be mapped to possible combinations of reserved, spot and on-demand instances available on the market.

The theoretical instance requirements will be mapped to service manipulation requests such as 'create a virtual machine' from a specific provider, which in turn will be mapped to low level request, such as an OCCI/Openstack 'Create VM' command.

Service information regarding provisioning of VM or listings of current VM will feed back into the pricing information.

## 7.4 Semantic Description of Pricing

The ontology to support the pricing of high level requirements is described in Figure 51 (below).

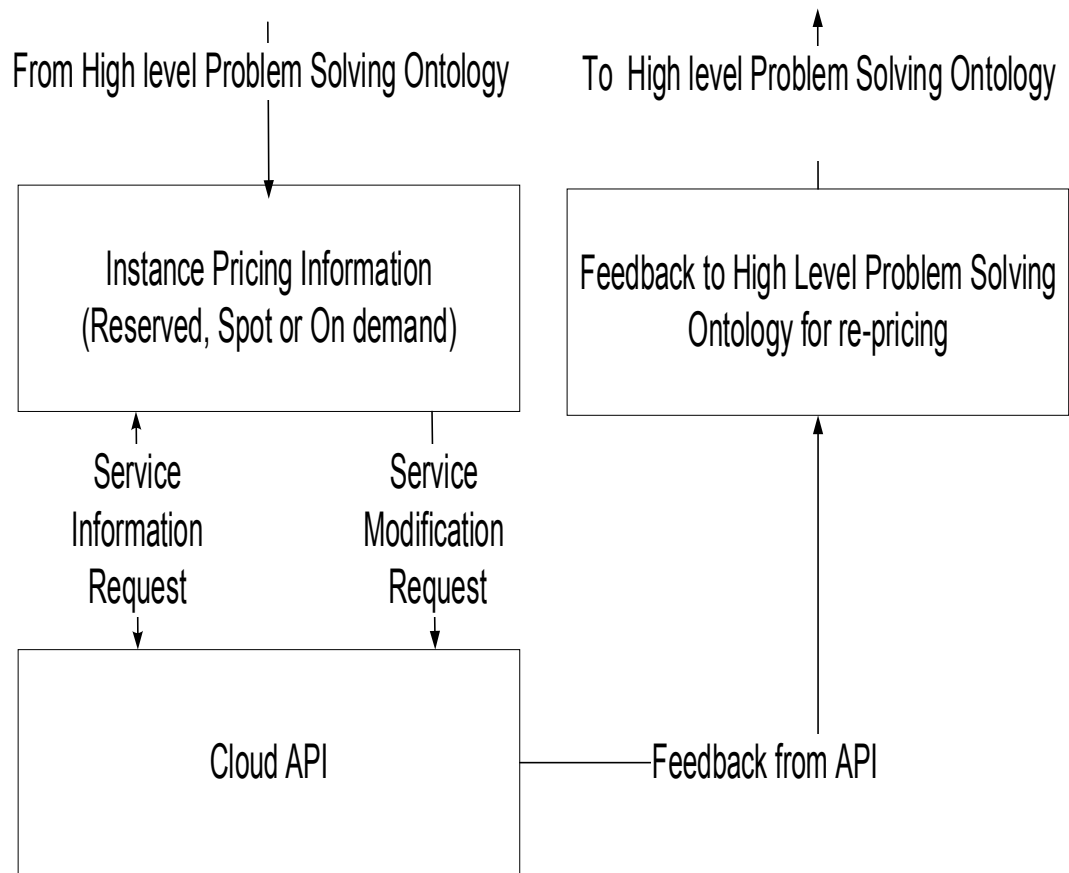


**Figure 51 - High Level Problem Solving Ontology for Pricing**

The full ontology design is shown in Appendix D.

The Figure 52 (below) shows how pricing information allows the user to trigger service requests to cloud API to change usage of cloud resources.

Information from cloud resources such as changes in prices will trigger repricing actions in the high level ontology.



**Figure 52 - Trigger service requests to cloud API**

## **7.5 Pricing Case Study**

### **7.5.1 Purpose**

The pricing case study demonstrates assertions (ABox entries) can be developed for the pricing ontology described in the chapter and shown in detail in Appendix D Hybrid Meta-heuristic Pricing Ontology.

### **7.5.2 System under Test**

Reusing the information from case study for the requirements semantic frame seen in Chapter 6 a pricing case study will be described using the theoretical pricing semantic framework. High level requirements are defined as tasks which bring together Problem Solving Methods (PSM) and knowledge domains. PSM CPU usage is described in a notional form such as 'ticks' which will be mapped to actual CPU usage in a payment structure.

Domains utilise storage. Once the payment structure is mapped a price can be generated. Actual usage statistics from cloud services will feedback to the semantic pricing framework, to adjust resource usages and the notional resource usage to actual resource usage mapping.

### **7.5.3 Application the of Proposed Method**

The concepts and relationships described in Figure 51 High Level Problem Solving Ontology for Pricing (above) and the ontology shown in Appendix D Hybrid Meta-heuristic Pricing Ontology was used to develop Assertions (ABox entries).



## 7.5.4 Results

### Tasks

The ABox entries for Tasks are shown in Figure 53 below.

```
Task (Find_References, PSM (Parser), Domain(Document_Collection))
Task (Format_References, PSM (Formatter), Domain(Refernce_System))
Task (Import_References, PSM (Importer), Domain(Refernce_System))
Task(Extract_Text, PSM(Text_Extractor), Domain(Software_Tools))
Task(Indexation, PSM(Indexor), Domain(Software_Tools))
Task(Match, PSM(Matcher), Domain(Software_Tools))
Task(Reporting, PSM(Reporter), Domain(Software_Tools))
```

**Figure 53 - Pricing Tasks ABox Entries**

### Usages in Notional Units

The resource unit ABox entries for Pricing are shown in Figure 54 below.

```
Has_PSM_Usage(Parser, CPU(20, Ticks, I3, Iteration), RAM(2, GB))
Has_PSM_Usage(Formatter, CPU(10, Ticks, I3, Iteration), RAM(1, GB))
Has_PSM_Usage(Importer, CPU(40, Ticks, I3, Iteration), RAM(1, GB))
Has_PSM_Usage(Extract_Text, CPU(100, Ticks, I3, Iteration), RAM(1, GB))
Has_PSM_Usage(Indexor, CPU(150, Ticks, I3, Iteration), RAM(2, GB))
Has_PSM_Usage(Matcher, CPU(200, Ticks, I3, Iteration), RAM(1, GB))
Has_PSM_Usage(Reporter, CPU(50, Ticks, I3, Iteration), RAM(2, GB),
Storage(2, GB), Network(0.25, Gbps))

Has_Domain_Usage(Document_Collection, Storage(3, GB))
Has_Domain_Usage(Reference_System, Storage(0.25, GB))
Has_Domain_Usage(Software_Tools, Storage(0.5, GB))
```

**Figure 54 - Pricing Resource Usage ABox**

The usage assertions can now map to a payment structure via a notional resource usage to actual resource usage mapping. The IBM payment structure that was described in the case study for CSA in chapter 5 is shown below.

### **IBM Payment Structure**

The ABox entries for the IBM Payment Structure is shown in Figure 55 below.

<pre>Payment_Structure(Standard_Capacity, CPU(96, Cores), RAM(384, GB), Storage(3.6), TB), Neworking(1, Gbps), Cost(\$7100, Month))  Payment_Structure(Enterprise_Capacity, CPU(192, Cores), RAM(384, GB), Storage(3.6), TB), Neworking(1, Gbps), Cost(\$7100, Month))</pre>
--

**Figure 55 - IBM Payment Structure ABox**

### **7.5.5 Evaluation**

The case study demonstrates how the theoretical semantic pricing framework can encode a real-world example. The usefulness of the problem solving approach is demonstrated by a task representing a requirement that brings together a PSM and a knowledge domain.

The PSM can be instrumented at notional level and then mapped to the target payment structure to generate a PSM price. The knowledge domain can mapped in a similar manner. This allows reuse of pricing information for PSM and knowledge domains, which is key to theoretical aspects of the semantic framework for pricing.

The case study highlights the requirement for notional instrumentation of PSM and for notional to payment structure mapping.

## 7.6 Comparison to Syntactical Pricing Frameworks

Yeo et al. [187] present a pricing model based on Quality of Service (QoS) based on (pre-paid) reserved instances, they use a number of simple calculations shown in Table 11 below:

Name	Configured Pricing Parameters
FixedMax	\$3/CPU/h
FixedMin	\$1/CPU/h
FixedTimeMax	\$1/CPU/h(12AM–12PM) \$3/CPU/h(12PM–12AM)
FixedTimeMin	\$1/CPU/h(12AM–12PM) \$2/CPU/h(12PM–12AM)
Libra+\$Max	\$1/CPU/h(PBasej), $\alpha=1,\beta=3$
Libra+\$Min	\$1/CPU/h(PBasej), $\alpha=1,\beta=1$

**Table 11 - Pricing for Processing Power Taken from Yeo et al.**

In the table PBase is a base price and,  $\alpha$  and  $\beta$  are static and dynamic components of the Libra model.

Compared to the semantic framework proposed in this thesis there are clearly a number of deficiencies. Firstly, in the proposed framework the common semantic description allows pricing to be related to Cloud Service Agreements and requirements, this provides the ability to relate pricing to larger number of variables beyond CPU hours and more importantly complex but still computable relationships expressed as logic. The complex relationships and knowledge can be adjusted as assertions are added to the ABox.

Secondly, the concept of notional units of CPU and other variables (Memory Usage, Transfer Time and Geographic Location) are expressed in the proposed framework. This allows better pricing decisions as a user of cloud services can compare services from a number of providers.

Thirdly, the proposed framework allow reasoning on the models created. This allows complex models to be created implicitly using modelling concepts such as inheritance. The reasoning process can be checked for correctness and reasoning can be explained to gain further insights into complex models.

The Flexprice model [180] uses a simple acyclic graph to represent a 'job' which is made up of 'tasks' which is submitted for pricing. The Flexprice model comprises a number of mathematical formulae which are used to price the job, to represent factors such as CPU and memory usage. The pricing model proposed in this thesis provides a highly sophisticated task model represented as Problem Solving Ontology (PSO) where the relationship between task elements and decomposition of tasks are expressed as complex semantic relationships. The concepts and relationships contained in the proposed framework can incorporate mathematical formulae but also incorporate complex logical constructs.

Martens et al. [188] provide a broader scope of cloud computing pricing, described in the list below to provide Total Cost of Ownership (TCO):

- Costs associated with the selection of Cloud Computing Services
- Evaluation of Selection of Service Provider
- Service Charge
- Implementation Configuration Integration and Migration Costs
- System Failure Costs

The costs listed above are simply summed to form TCO. The pricing model proposed in this thesis provides a superior approach as the variable costs associated with the selection and evaluation costs associated with service and provider selection are eliminated by the ability to query model generated by collaborative ontology development. The ABox created will expand over time leading to a greater range of pricing situations being considered.

The pricing model has sophisticated concept and relationship representations that go beyond simple summation.

## **7.7 Discussion and Conclusions**

Pricing of cloud services have previously used simplistic 'syntactical' or mathematical models. Using semantic description provides an approach that can price cloud computing services in terms of concepts and relationships. These concepts can relationships can be reasoned across to provide new knowledge and has the ability to describe many pricing combinations without the need to explicitly describe all the combinations.

Each cloud service provider has a number of pricing combinations. Although, the combinations are semantically equivalent in describing CPU power, memory, storage and network characteristics, they express the service characteristics in syntactically different ways. It is difficult to compare and price services without semantic description. The rise of on-demand instances or spot (auction) markets require pricing decisions to be made quickly and constantly work on new service descriptions fed from cloud service providers. Semantic description can deal with changes in service description quickly by rearranging semantics rather than reprocessing large combinations of service descriptions.

A novel workflow and semantic framework design were presented along with a case study to demonstrate the approach.

## **Chapter 8      Semantic    Description    for    Benefits Management of Cloud Computing**

### **8.1 Chapter Overview**

A large amount of resources are being invested in cloud computing services. It is important to consider the value of these investments.

An existing Benefits Management approach was enhanced, using semantic techniques and was tested by encoding a number of existing case studies into the newly developed framework.

A new case study was developed to test the efficacy of the newly developed framework.

### **8.2 Introduction**

Cloud computing is a relatively new IS/IT technology which many organisations are beginning to use and are considering investing in. Organisations may not have considered how such investments will deliver business benefits. Benefits management in cloud computing has been examined, by using a number of primary and secondary case studies, to provide a unique contribution in this area. The benefits management approach is used to develop a semantic description to structure the knowledge gained from the case studies.

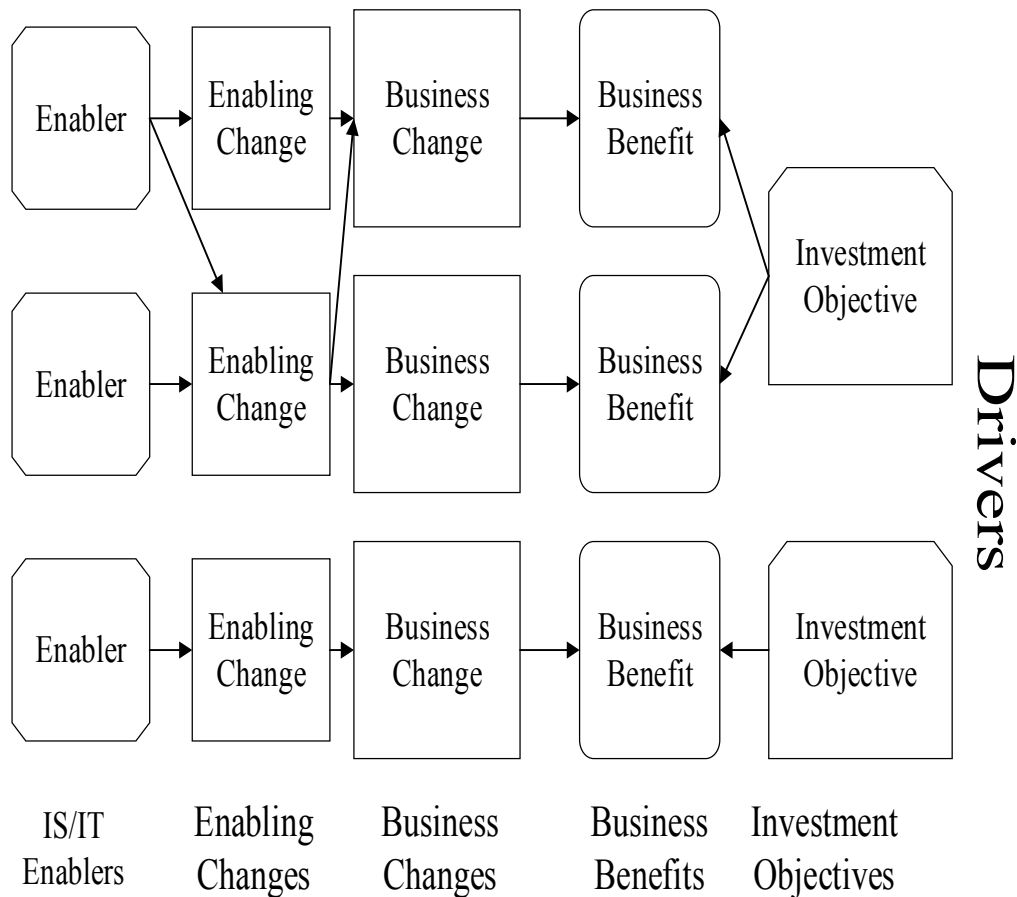
The motivation for using an semantic description is to abstract knowledge and reasoning from the knowledge domain being modelled, in order to develop a number of reasoning approaches based on attributes such as organisation size, type of cloud technology used and other factors to provide clear, precise and unambiguous definitions of the benefits derived from

cloud computing. A detailed discussion of the advantages of reasoning is made in Section 8.10.

Future work will see the expansion of the semantic description to allow multiple stakeholders to add further case studies. This will provide additional reasoning and scenarios to the semantic description.

The 'Benefits Management' technique is described and enhanced through the introduction of a semantic benefits management framework and is used to develop an semantic description from a number of case studies, which can be accessed and edited by other researchers. The semantic description developed for Benefits Management allows a service to be created, which can be accessed and enhanced by multiple stakeholders.

The Benefits Management approach attempts to link IS/IT enablers such as new technology advances to create change in the organisation. The changes are termed enabling changes. IS/IT enablers are only useful if they enable change in the organisation. Enabling changes trigger business changes in the organisation that delivers benefits. The benefits meet clearly defined investment objectives. The Benefits Management process is encapsulated in the Benefits Dependency Network (BDN) shown in Figure 56 below.



**Figure 56 - Benefits Dependency Network**

A holistic approach should be taken in the Benefits Management approach. Intangible benefits should not be ignored. Previous approaches to examining the value delivered from IS/IT investment have concentrated on financial measures and may have ignored the full spectrum of benefits.

The classification of benefits is the next stage of the approach. A business case is presented to key stakeholders who will then make investment decisions. The more explicitly a benefit can be expressed the easier it is to gain commitment to investment. A benefit expressed in financial terms will more easily gain acceptance than a benefit that is merely observable. Benefits can also be classified in terms of the next action for a given benefit, if something new should be done, continued or stopped.



The final stage of the Benefits Management process is to identify benefits into the types. High potential investments may deliver high value but carry high risk. Strategic investments are central to the success of the business. Key operational investments can be improved to increase productivity in the business. Support investments deliver the least value to the business and may be stopped if they become more expensive.

An issue with the current Benefits Management process is that it does not express the elements of the process (such as enablers, changes and benefits) and the relationships between the elements of the process in terms of semantics. A key is to use a semantic framework to improve the knowledge representation within the Benefits Management process. New semantic description tools such as Web Protégé [159] allow multiple stakeholders to build a Benefits Management semantic description through collaborative developments.

In the next section IS/IT enablers seen in cloud computing will be considered.

### **8.3 Cloud Computing Enablers**

A number of IS/IT enabling technologies have been identified in cloud computing. Provision models such Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) have been defined by NIST [13]. IaaS is the lowest level of enabler, where users procure hardware and operating system resources at a cost. PaaS brings together infrastructure, programming languages and data storage in a single package. SaaS provides customers with the ability to rent software packages on demand. Ownership models of cloud resources have also been defined by NIST [13] as public clouds, private clouds and hybrid clouds. Public clouds are provided by third parties at an agreed service level and price. Private clouds use cloud technology to provide services to customers

within an organisation. Hybrid clouds use both public and private clouds to provide services to customers.

Technologies built on cloud computing such as Big Data [189], Data Science [190] and storage services [191] are key enablers for generating change and benefits in cloud computing investments.

A number of enablers have been identified from secondary sources [22][192][193][172][194][195][196] which are now discussed. Cost is a primary enabler, IaaS provides low cost of ownership and the ability to manage cost. Ease of movement from test to production is facilitated by allowing a number of virtual instances can be procured and used to move from test to production. Large scale storage with low cost of ownership is provided by storage that can be purchased on demand and is managed and backed-up in the cloud.

Alternative ways of working and new products are being created by cloud computing. Shared development spaces between organisations especially in public clouds, can provide joint developments or provide greater customer intimacy. Organisations can create new products, especially on the PaaS platform. Data Science, Big Data and 'Smart Cities' [197] become feasible for small and medium size organisations. Flexibility of resources allow organisations can downsize/upscale on demand.

New markets and marketing can be accessed. The marketing power of cloud computing allow cloud solutions to be marketing tools, with an organisation's status improved by having a cloud computing solution. Many large corporations and government organisations require solutions to be cloud based, for example the United Kingdom's G-Cloud [198].

Private and public organisations can offer infrastructure and services to other organisations, to reduce ownership cost or to generate revenue. Public

organisations can create cloud infrastructure for economic development [199].

There are a number of operational enablers in cloud computing adoption. Cloud storage and infrastructure solutions can be used to manage disaster recovery [200]. Infrastructure management tasks can be reduced, which allows employees to concentrate on more skilled work or to develop new skills. Cloud services can be delivered to a number of devices [201]. The security of the infrastructure can be improved [202].

## 8.4 Cloud Computing Case Studies

A number of case studies were developed from primary and secondary sources. The case studies deal with different aspects of cloud computing, as described in the Table 12 below.

Organisation	Type	Description
Organisation A [22]	Micro Start-up Company	Provides solutions to the music promotion industry using PaaS/ public clouds.
Organisation B [22]	Actuarial Services Consultancy	Supplier of economic modelling reports using IaaS/PaaS on public/private clouds.
Organisation C [22]	Public Sector Division of Large Software Company	IaaS and SaaS solutions via private clouds
Organisation D [22]	Public Sector Managed Services organisation	Shared service between two local authorities using IaaS/SaaS.
Organisation E (new primary)	A large local authority	Adoption of IaaS in a large local authority with a commercial partner.

research)		
Organisation F [192]	Oil and gas company migration to IaaS	Migration from an in house data centre.
Organisation G [193]	Media Group	Software as a service for distributed media workers.
Organisation H [172]	Quality of service for three cloud services	Study of factors effecting ranking of quality of service in IaaS.
Organisation I [194]	University's adoption of cloud technology	Cloud adoption in an educational context
Organisation J [195]	Security benefits in cloud computing	Identification of security benefits in cloud computing.
Organisation K [196]	Implementation of cloud computing by doctors in South Africa	The benefits of using cloud computing to enable benefits such as better communication.

**Table 12 - Organisations Reviewed**

The methodology used was to extract Benefits Management information from each case study which was then used to build an upper semantic description for Benefits Management. The Table 13 (below) shows the IS/IT enablers for each case study.

Enabler	Case Study
Cost	A-K
Ease of movement from test to production	A, B
Large scale storage with low cost of ownership	A, B, G
Shared development space	A, B
New products	A, B, E
Flexibility of resources	A, B, C, H, I
Marketing power of cloud computing	B, C
New markets and procurement models	C, D, E
Provide services to 3 <sup>rd</sup> parties	C, D, E
Create infrastructure for start-up companies	E
Disaster recovery	E
Device independence and geographical distribution	G, I, K
Improve employee satisfaction	F, H
Improved security	E, J

**Table 13 - Enablers cross-referenced to case studies**

The Table 13 (above) shows that many of the enablers were present in the organisations covered by the case studies. The enablers can be split into two groups, business and operational.

Cost was an enabler in all organisations. The usage of IaaS was seen as an enabler to reduce costs in the short-term and a major reason for the uptake of cloud computing. Repeated cost reduction may not be feasible in the long-term and other enablers should be examined.

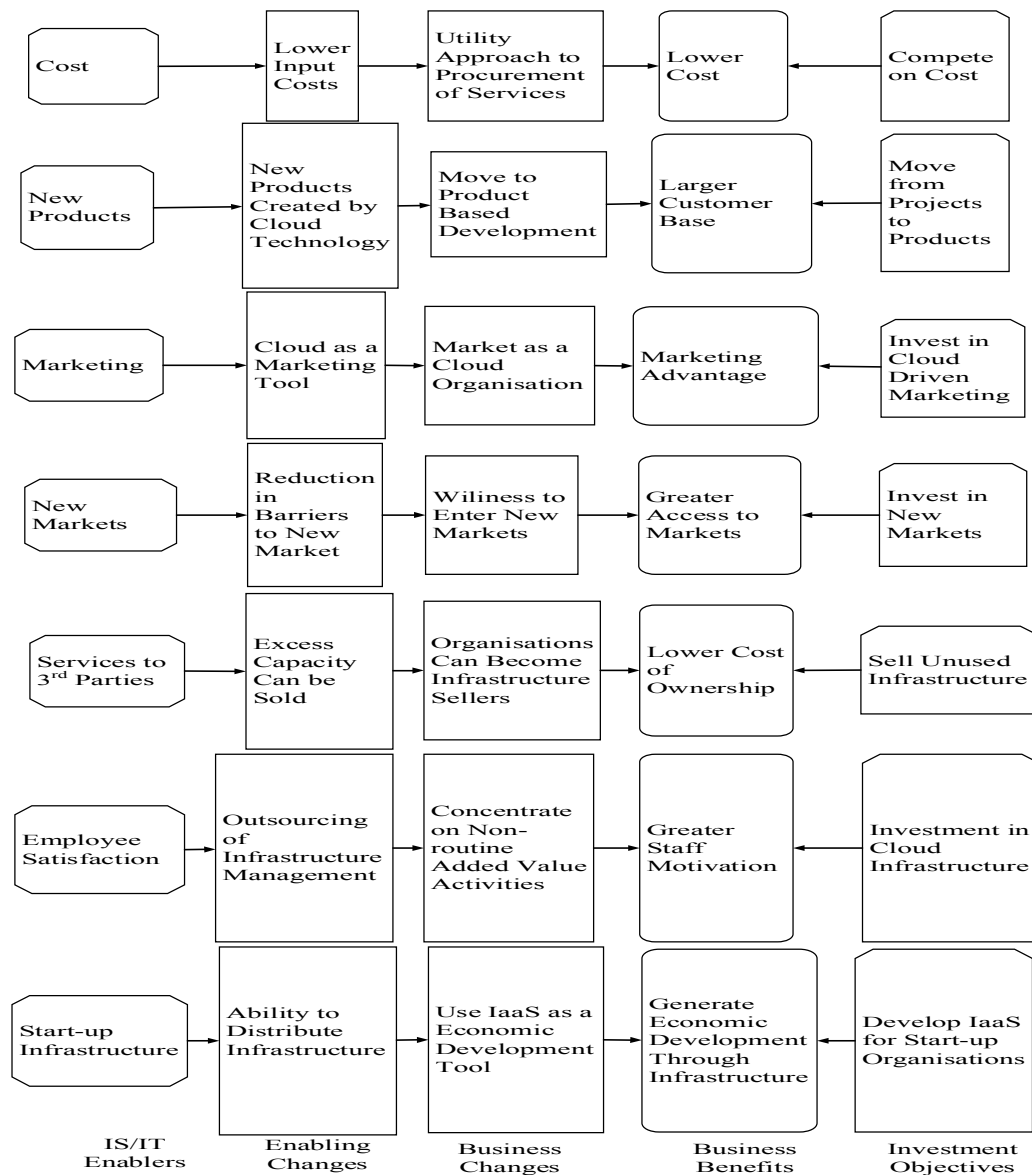
There are a number of new products being created by cloud computing such as storage solutions, data science applications and development environments. These enable organisations to gain new customers and to enter new markets.

Marketing cloud service enablers allow organisations to attract new customers and to maintain existing customers who may move to cloud based solutions in the future.

Lower costs of market entry are afforded by cloud computing which utilises rental of resources. Organisations can enter new markets without capital expenditure and maintenance costs. Organisations with existing infrastructure or those who require high levels of fixed resources can sell excess capacity.

The freeing up of staff from repetitive and tedious infrastructure development and maintenance is one of the main benefits. The dis-benefit of redundancies from outsourcing to the cloud is acknowledged. Organisations D & E in the case studies are large local authorities which have successfully adopted cloud infrastructure and redeployed staff into new customer facing roles.

Public authorities, academic institutions and non-profit organisations can use cloud infrastructure to allow start-up organisations to develop. Organisation E has used this approach to generate economic development The BDN for business enablers is shown in Figure 57 below.



**Figure 57 - Business Enablers**

The operational enablers are now described. The large-scale storage offered by cloud computing is a major operational enabler [43]. The backup, replication and disaster recovery of large amounts of data can be outsourced at a very low cost. Many organisations described in the case studies have large amounts of critical business data which is being moved into the cloud [203]. When low cost storage is combined with fast Internet connections an enabling cloud technology is created.

Resource procurement of hardware and software was previously a capital investment decision, requiring long-term planning, without the ability to adjust resources quickly as business needs change. The advent of cloud computing has seen the ability to purchase resources on-demand, through spot instances as well as through fixed resources to cope with base demand.

New approaches to the development of software solutions have been established using hybrid and public cloud technology. Organisation A has established a joint development environment with customers with a public cloud based platform. This has produced an operational approach that is more intimate with the customer and reduces operational risk through shared developments and cost.

Public and hybrid clouds enable organisations to create and store virtual machines at a low cost. Separate physical hardware and software is no longer required. Virtual machines can be moved from test to development more easily.

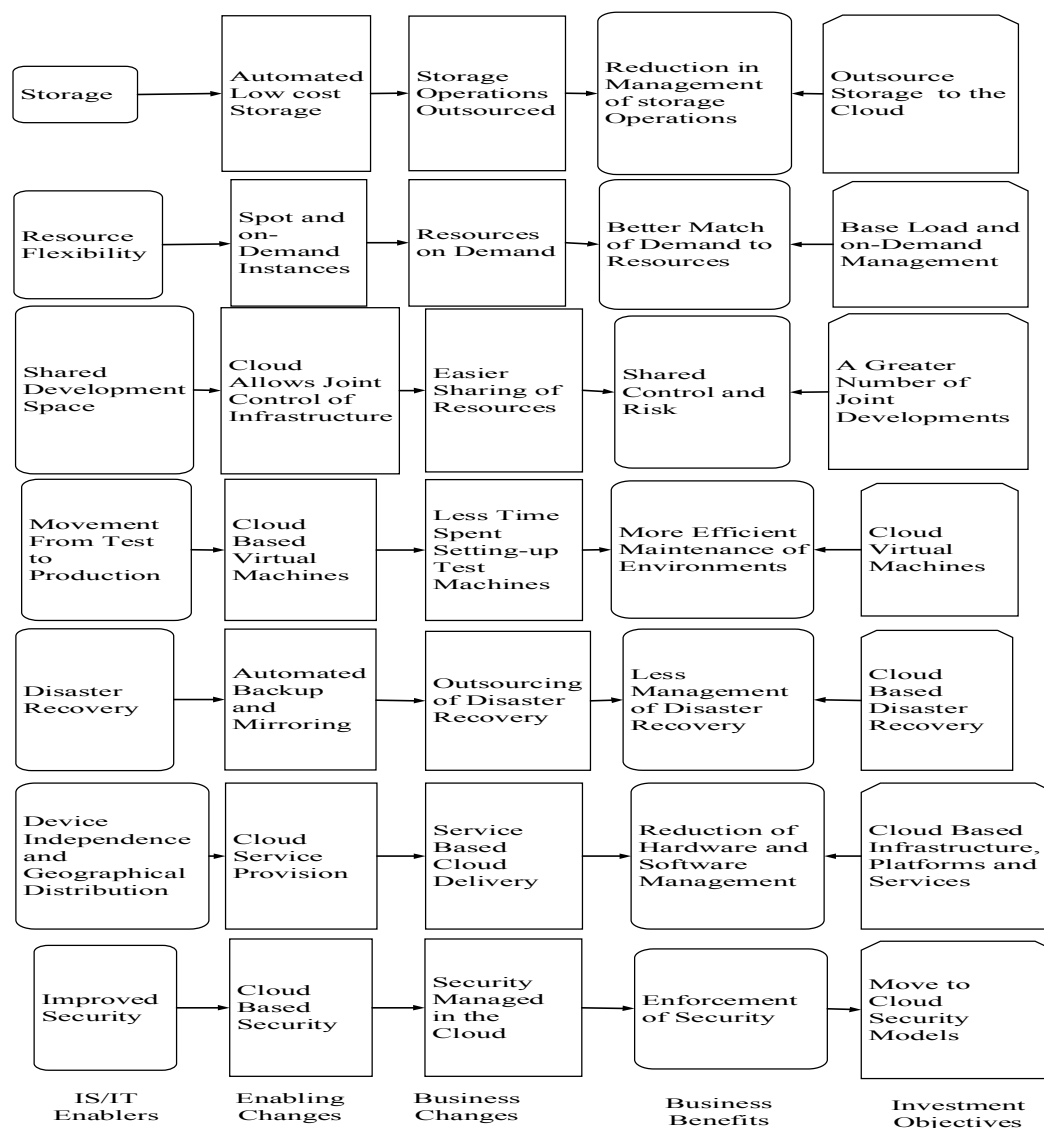
The provision of disaster recovery is an emerging market for cloud computing providers. Organisations will effectively outsource their disaster recovery operations to the cloud provider. This is advantageous because cloud storage is replicated and backed up multiple times across a number of geographical locations [204]. Virtual machines can be made ready to provide instant services if a company's own data centre is unavailable. Expertise can be concentrated at cloud providers that would be difficult to replicate outside large IS/IT providers.

Services can be accessed from a number of devices such as phone apps, tablets and desktop machines more easily using cloud based services [201]. The operational requirement to install and manage software and data falls on the cloud provider.



The high availability of data and secure access can be managed by the cloud provider. Systems and expertise will be more advanced than that afforded by small in-house providers. However, there are problems with outsourcing security due to loss of control of the organisation and conflict of interests if the cloud provider provides services to competitors.

The operational enablers for cloud computing are shown in the Figure 58 below.



**Figure 58 - Operational Enablers**

The benefits are classified in the Table 14 below. The financial benefits are centered on the lower cost of ownership from using utility infrastructure. New markets (such as government provision platforms) could be entered which would provide financial benefits. Operational efficiencies provide further financial benefits such as the ability to create new environments and to outsource the management of computing resources. Reduced fixed costs will result from the move to a 'rental' model as opposed to spending money on internal IS/IT infrastructure.

Quantifiable benefits include improvements in service quality, with the ability of users to vary the amount of resources they use. The speed of functionality delivery and the availability of resources were improved. There may be internal staff reductions due to cloud computing infrastructure investments. The operational benefits of the cloud based IS/IT such as the lowering of e-mail traffic and increased security in the cloud are measureable. Future benefits from new technologies seen in PaaS and enabling data science innovations can be measured using forecasting techniques.

The marketing benefits of cloud computing are important to many of the organisations. These benefits are difficult to measure in the short-term but are observable in internal and external marketing positions in the organisations.

Degree of Explicitness	Do New Things	Do Things Better	Stop Doing Things
Financial	Lower cost of ownership Reduced fixed costs.	Reducing time to create infrastructure	Managing own infrastructure. Grid computing

Quantifiable	<p>Improved quality of service</p> <p>Customer self service</p>	<p>Faster turnaround of new functionality</p> <p>Speed of delivery</p> <p>Availability improvements</p>	<p>Internal infrastructure</p> <p>Direct employment of staff through infrastructure outsourcing</p>
Measurable	<p>Lower e-mail traffic</p> <p>New markets for Big Data and Data Science</p>	<p>PaaS innovations</p> <p>Security of data.</p> <p>Improve customer satisfaction</p>	<p>E-mail traffic</p> <p>Storing information on individual computers</p>
Observable	<p>Better customer intimacy.</p> <p>Improved marketing</p> <p>Move from project to product based solutions</p>	<p>Actively market to customers</p>	<p>Waiting for customers to 'come to the organisation'</p>

	Sell infrastructur e and services outside the organisation		
--	---	--	--

**Table 14 - Classification of Benefits**

The cloud investment portfolio is shown in Table 15 (below). The portfolio shows long-term strategic investments for organisations adopting cloud technologies such as infrastructure, services and storage. Private clouds are being developed and there is some development of hybrid technologies which utilise combinations of private and public cloud ownership.

High potential investments are riskier investments that may yield higher returns. Small innovative organisations may use PaaS to deliver unique products that will differentiate them from the mass market. Data science investments enabled by cloud computing promise high growth, but may be high risk due to the immaturity of the technology in this area.

Key operational investments will be supported in the short- to medium-term. Private clouds will be developed by organisations at high cost to organisations, based on in-house servers or on customers' hardware. Non-cloud and grid computing solutions will be supported in the short-term but will be replaced by cloud technologies due to cost and usability issues. Public clouds will be important in the short-term for many organisations; however, their ubiquity and low cost will not generate competitive advantage in the long-term.

Strategic	High Potential
IaaS SaaS Cloud storage Private/hybrid clouds	PaaS Marketing of Cloud Computing Market places such as G-Cloud Big Data & Data Science Shared Services
Existing customers with their own hardware Clustered in-house servers with cloud extensions Private Cloud	Non-cloud Based Software Grid Computing Public Cloud (long-term)
Key Operation	Support

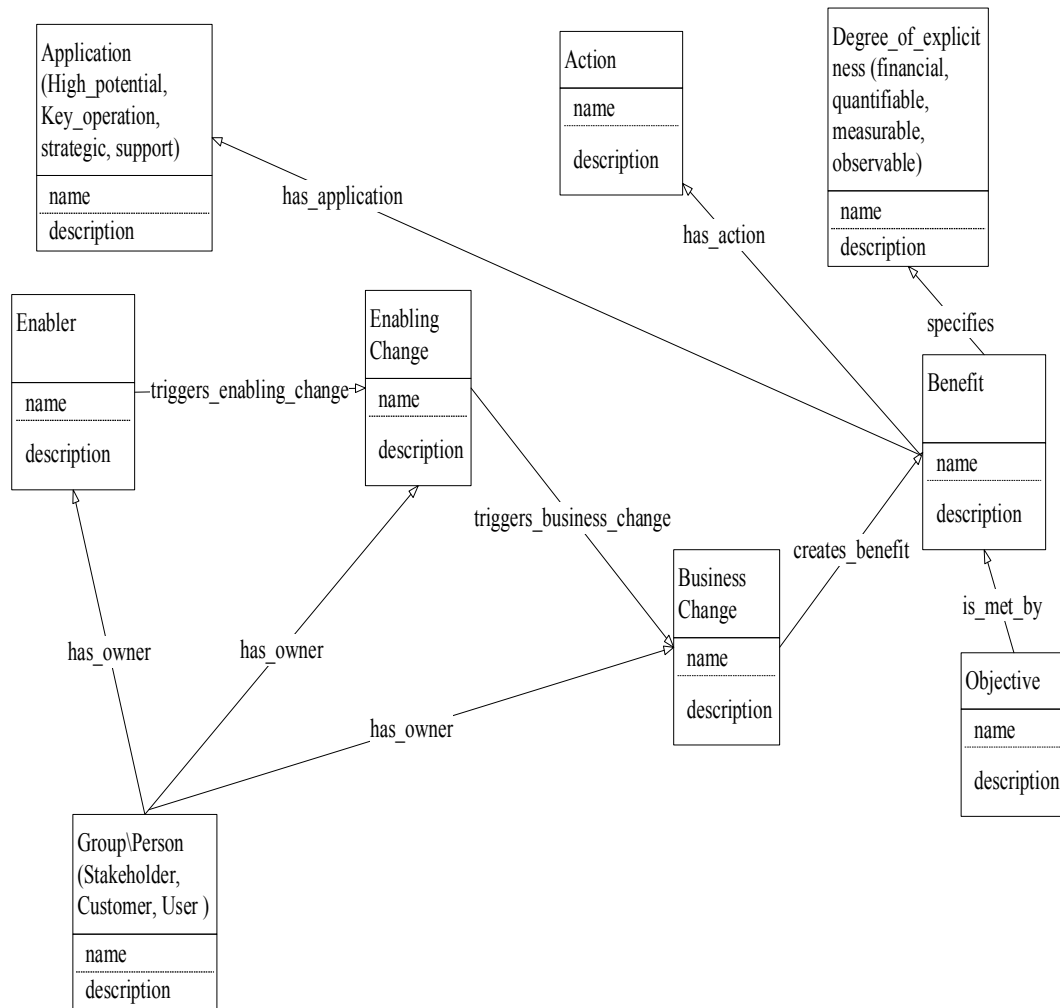
**Table 15 - Cloud Investment Portfolio**

## **8.5 A Semantic Description for Benefits Management in Cloud Computing**

A semantic description was generated from case studies previously described. This provides a formal description of the Benefits Management terminology, relationships and assertions provided by the case studies.

The semantic description was created so that the terminology can be reused across a number of projects. The terminology for the benefits management semantic description has been uploaded to the WebProtégé website [205]. This allows the full semantic description to be viewed, critiqued and used by other researchers. The assertions for the case studies described are held in a separate semantic description file that can be supplied or uploaded on request. Also, the assertions can be overlaid on the terminology to provide a full semantic description. The decision to separate the terminology and assertions was to allow for the reuse of the terminology.

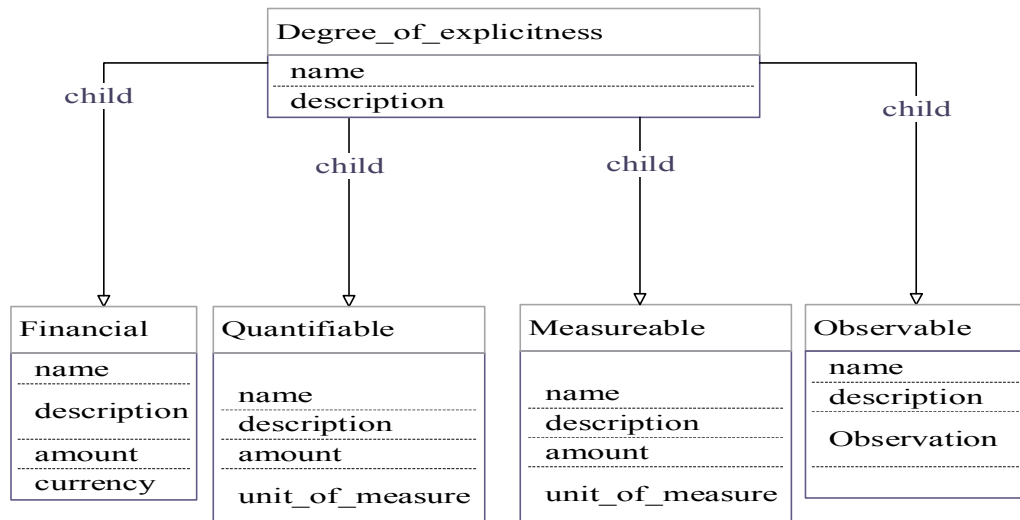
## 8.6 Semantic Description Classes



**Figure 59 - Overview of Main Semantic Description Concepts**

The Figure 59 (above) shows the main semantic description concepts. A full description can be found in the WebProtégé project [205]. The Benefits Dependency Network forms the core of the semantic description with semantic linkages between enablers, changes and benefits. Each entity can be related to the owner such as a group or stakeholder. Benefits can be linked to objectives and be classified or related to investment portfolio applications.

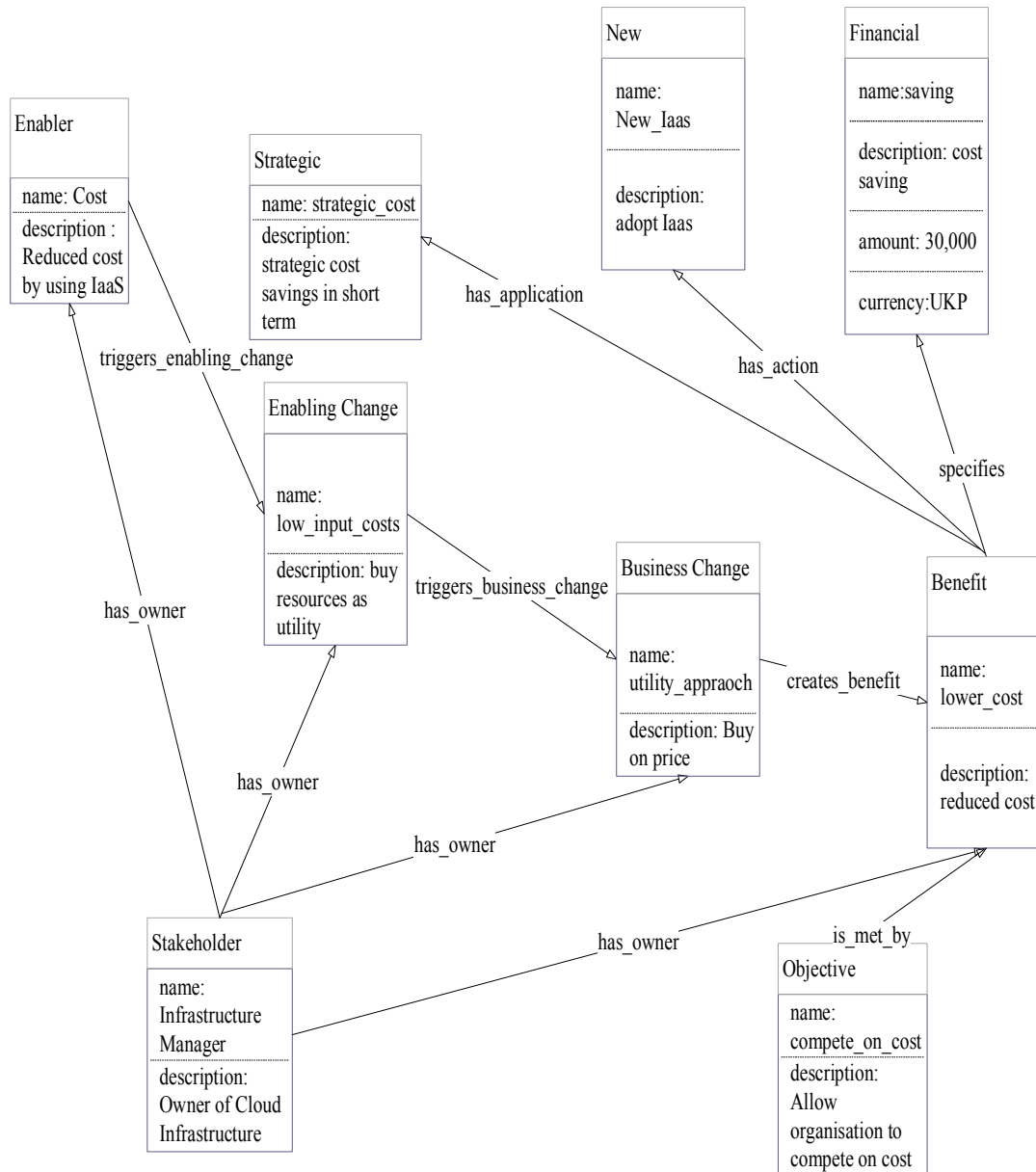
The names entities shown in parenthesis are child entities. An expanded example for 'Degree\_of\_explicitness' is shown in Figure 60 below.



**Figure 60 - Overview of Main Semantic Description Concepts**

## 8.7 Implementation

An example of the semantic description class implementation (assertions) for Benefits Management is shown in Figure 61 below. The cost enabler sees lower input costs in the business. The organisation purchases on price. Cloud computing resources are treated as utilities which can be supplied by a large number of suppliers. This gives the benefit of lower cost to the business which, meets the business objective of competing on cost. This is a new strategic investment which can be expressed financially.



**Figure 61 – Semantic Framework Implementation for Benefits Management**

## 8.8 SPARQL Queries

SPARQL [133] can be used to provide Benefits Management outputs from the semantic description. The namespace prefix 'bm' signifies 'benefits management'. SPARQL traverses the semantic data held in the semantic description to produce outputs.



Three examples of useful outputs from the Benefits Management approach identified in the literature which are represented as SPARQL queries are shown in the Table 16 below. The 'Benefits Stream' query traverses the Benefits Dependency Network (BDN) to describe the linkage between enablers, change, benefits and objectives. The 'Stakeholder Analysis' query examines the relationships between benefits and their owners and the stakeholders' commitment to the benefits. The 'Dimensions of Competence' query examines the relationship between drivers in the business such as the need to reduce costs in the business and the ability to meet the drivers from competences within the business.

Description	SPARQL Query
Benefits Stream [4] p. 102 - A set of related benefits and their associated business and enabling changes and enabling IS/IT	<pre> SELECT    ?enabler    ?enablingchange ?businesschange    ?benefit    ?objective ?action ?degree_of_explicitness WHERE {     ?enabler    bm:triggers_enabling_change ?enablingchange.     ?enablingchange bm:triggers_business_change ?businesschange.     ?businesschange    bm:creates_benefit ?benefit.     ?objective bm:is_met_by ?benefit.     ?benefit bm:has_action ?action.     ?benefit    bm:specifies ?degree_of_explicitness } </pre>
Stakeholder analysis [4] p. 179 – Stakeholder groups, their benefits, changes and	<pre> SELECT    ?owner    ?benefit    ?change ?commitment ?commitment_action WHERE { </pre>

commitments to change	<pre> ?benefit bm:has_owner ?owner. ?benefit bm:needs_change ?change. ?change                bm:has_commitment ?commitment. ?commitment bm:has_commitment_action ?commitment_action } </pre>
Dimensions of competence [4] p. 114 – The different capabilities of the organisation (this will get competency type and description of competency)	<pre> SELECT  ?driver  ?competence  ?type ?description WHERE {     ?driver                bm:has_competence ?competence.     ?competence a ?class.     ?class rdfs:label ?type.     ?competence                bm:description ?description } </pre>

**Table 16 - Benefits Satisfied by SPARQL Queries**

## **8.9 Benefits Management Case Study**

### **8.9.1 Purpose**

A number of case studies from papers were used to develop the semantic framework for benefits management. The case studies are outlined in section 8.4. A new case study was developed from primary sources to demonstrate semantic framework generated from this research.

A case study of unpublished work of Organisation E described in section 8.4 Cloud Computing Case Studies (above) will now be described.

### **8.9.2 System under Test**

Organisation E is a local government organisation and, a leader in the adoption of cloud computing services in the United Kingdom's public sector. The enablers for organisation E have been described in the "Enablers cross-referenced to case studies" table previously described (above). The semantic framework for benefits management encoding for organisation E will be not be described.

### **8.9.3 Proposed Method**

The concepts and relationships described in Figure 61 Semantic Framework Implementation for Benefits Management (above) and Appendix C An Upper Ontology for Benefits Management of Cloud Computing Investments were used to encode assertions (ABox entries).

### **8.9.4 Results**

The encoding of the ABox as a series of triples is shown in Figure 62 (below).

```
Stakeholder(Management, Enabler(Cost, Enabling_Change(Lower_Cost),  
Business_Change(Low_Cost_IT)))  
  
Stakeholder(Workforce, Enabler(New_Products, ,  
Enabling_Change(Cloud_Computing), Business_Change(Innovation)))  
  
Stakeholder(Management, Enabler(New_Markets_and Procurement_Models,  
Enabling_Change(Cloud_Procurement_Models),  
Business_Change(Rental_On_Demand_Procurement)))  
  
Stakeholder(Management, Enabler(Provide_Services_to_3rd_Parties,  
Enabling_Change(Cloud_Reseller),  
Business_Change(Incomming_Revenue)))
```

```

Stakeholder(IT_Department, Enabler(Disaster_Recovery),
Enabling_Change(Cloud_Infrastructure),
Business_Change(Outsource_Recovery)))

Stakeholder(IT_Department, Enabler(Improved_Security),
Enabling_Change(Cloud_Infrastructure),
Business_Change(Outsource_Security)))

Business_Change(Low_Cost_IT, Benefit(Low_Cost), Degree(Financial,
Quantifiable, Measureable, Observable), Objective(Reduce_Costs),
Action(Buy_Least_Cost_Services))

Business_Change (Innovation, Benefit (New_Products), Degree
(Observable), Objective (Examine_New_Products), Action
(Perform_Analysis))

Business_Change (Rental_On_Demand_Procurement, Benefit (Low_Cost),
Degree (Financial, Quantifiable, Measureable, Observable), Objective
(Reduce_Costs), Action (Buy_Least_Cost_Services))

Business_Change (Incomming_Revenue, Benefit (Inflow_of_Income),
Degree (Financial, Quantifiable, Measureable, Observable), Objective
(Resell_Cloud_Services), Action (Setup_Reseller_Company))

Business_Change (Outsource_Recovery, Benefit (Lower_Costs), Degree
(Financial, Quantifiable, Measureable, Observable), Objective
(Cloud_Based_Recovery), Action (Use_Cloud))

Business_Change (Outsource_Security, Benefit (Lower_Costs), Degree
(Financial, Quantifiable, Measureable, Observable), Objective
(Cloud_Based_Security), Action (Use_Cloud))

```

**Figure 62 - Benefits Management Abox Encoding**

### 8.9.5 Evaluation

The encoding of benefits management into the semantic framework shows the theoretical design of the semantic framework can be used to represent a real-world case study.

A processes is seen in the semantic framework guide the encoding of the case study from stakeholders with enablers through to objectives and

actions, this provides a new theoretical workflow for researchers and practitioners to follow.

A number of useful semantic patterns can be seen around low costs and procurement of low cost services.

Concepts in the benefits management semantic framework can be mapped to CSA, requirements and pricing semantic framework concepts.

## **8.10 Advantages of Reasoning**

The use of reasoning and reasoning software such as Pellet [95] has a number of benefits. Firstly, the ontology developed can be checked for logical correctness. A main driver for development of the semantic framework is to encode cloud computing service descriptions so a formal model is created to reduce ambiguity and create information that is amenable to logical processing by computerised systems. In the Benefits Management case study reasoning can check if the logical relationships between classes such as Enabler, Enabling Change, Business Change and Benefit are logically correct. It is also possible to check the developed semantic framework for inconsistency, where classes (for example the class Enabler) in semantic framework has no instances and therefore the model should be correct. This iterative debugging and collaborative ontology development approach is key to the success of the semantic framework being developed. The Benefits Management information (as with Cloud Service Agreements, Requirements and Pricing) can be seen to logically correct and useable in a computerised system. This was not the case with the original Benefits Management approach seen in Ward and Daniel [4], where the relationships between elements of the Benefits Management may or may not be logically correct and may or may not be consistent.

A second advantage of reasoning is allowing users to comprehend entailment. Entailment describes the relationships present when one

statement occurs as a logical consequence of another. Reasoning produces a chain of entailment that can be tested and audited. If an unintended consequence occurs this can be debugged more easily. In a real-world deployment of the semantic framework there would be thousands 'individuals', for example Enablers, Enabling Changes, Business Changes and Benefits. Automated reasoning is required to test and debug such as deployment.

The third advantage of reasoning is the ease of modelling the semantic framework. The assertion of hierarchy (inheritance) produced by reasoning means the semantic framework does not require inheritance is explicitly stated throughout the framework, but is achieved by logical description and reasoning.

## **8.11 Comparison to other Business Benefits Frameworks for Cloud Computing**

The issues seen in business frameworks for cloud computing are similar to those seen in cloud computing frameworks,

Weinhardt et al. [206] presents a business model that is represented using "syntactical" representations a simple entity-relationship hierarchy is presented to present "applications" that map to storage, compute or software as a service resources. There is no mapping from low level descriptions at an infrastructure level to high level requirements. There is an emphasis on low level requirements, concentrating on machine resources. There is no attempt to examine the business benefits generated by cloud computing investments.

Marston et al. [207] provides a paper that similar to many high level business papers. There are some broad definitions of cloud computing concepts. The business benefits analysis comprises a simple Strengths, Weaknesses, Opportunities Threats (SWOT) matrix. There is no attempt to build a

sophisticated analysis model with requirements and pricing linked to business benefits, as with the framework proposed in this thesis.

## **8.12 Discussion and Conclusions**

The benefits derived from cloud computing investments have been confined to articles in business journals or magazines or, economic discussions.

Two novel pieces of work were presented that provide substantial contributions to knowledge in this area are listed below:

- The semantic description of an existing Benefits Management approach
- A number of case studies were codified in a semantic representation and then as description logic based ontology

The existing Benefits Management approach has been used successfully in implementation of information systems. However, it was originally a syntactical approach, comprising simple diagrams with lines linking textual descriptions of the concept. The approach was analysed to extract the semantic descriptions of the various aspects of the benefits management approach which makes it a more powerful technique by providing enhanced descriptive ability, this provides a number of benefits, which are listed below:

- The ability to describe concepts and relationships semantically
- Description logic ontology can be developed from the semantic representations which can reasoned across, searched and shared between contributors
- The ability to merge with other semantic descriptions, such as Cloud Service Agreements (CSA) requirements and pricing

A number of case studies were encoded into the semantic description framework to prove the approach and to provide ontology that can be used by other researchers and practioners.



## **Chapter 9      Conclusions and Future Work**

### **9.1 Chapter Overview**

The proposed framework is summarised and critically analysed. The proposed framework is then compared to other existing frameworks to assess its usefulness.

Future work sees further development of the proposed framework moving towards a 'Unified Semantic Framework' for cloud service description.

### **9.2 Introduction**

The main deliverables from the research undertaken was the development of new and unique semantic framework for description and specification of cloud services. It was possible to create specifications for Cloud Service Agreements and link the agreements to requirements and pricing specifications and, to consider these three elements in the context of a business benefits delivered.

A number of case studies were developed to prove the efficacy of the approach and to implement the semantic framework as ontology. The results of the research were presented as peer reviewed papers in journals and conferences and, as implementations of the semantic framework delivered as ontology on online ontology platforms.

The chapter will continue in six sections. Firstly a discussion of results from each of the four components of the proposed framework. Secondly, a discussion of the evaluation of research carried out. Thirdly, a critical analysis of the approach is carried out, considering the strengths and weaknesses of the approach. Fourthly, a summary of the review of other

frameworks. Then there are conclusions and a discussion for the main contributions delivered by the research. Finally, there is a discussion of future research and applicability of the framework to emerging technologies, which are related to cloud service description.

### **9.3 Discussion of Results**

The proposed framework has four elements (agreements, requirements, pricing and Benefits management). The results gathered for ontology developed for each element and connections between the elements is now considered.

The Cloud Service Agreements showed detailed ontology encoding for an actual CSA for Oracle and IBM cloud services. The encoding showed that both CSA were semantically similar and payment structures, suspension information, terms and conditions, risks and acceptable policy being modelled. There is a large overlap with pricing descriptions and requirements.

The requirements ontology used a Problem Solving Ontology (PSO). The requirements were mapped to a unique two tier concept/relationship framework. It was demonstrated a real world problem workflow could be represented in the proposed framework.

The results from the pricing built on the results from requirements case study. This demonstrates the elements of the proposed framework can be linked using the semantic encoding approach of PSO. The concept of notional units for CPU, memory and disk usage was introduced.

The Benefits Management results demonstrated that a large number of case studies could be encoded into the semantic framework. The ability to query the framework was demonstrated by a number of SPARQL queries.

In conclusion, the results show the individual elements of the framework can be encoded into the framework and there are strong linkages between concepts, relationships and assertions in the proposed framework.

## **9.4 Discussion of Evaluation**

The Cloud Service Agreements (CSA) for two companies were modelled using the semantic framework. It was possible to model a large part of the CSA. However, a major issue was highlighted in the fact that building the assertions for the semantic framework was very labour intensive. To model all CSA available would take several person years. This highlights the need for collaborative ontology development.

The requirements for a real world case study were modelled successfully. The Problem Solving Ontology was seen as being useful, with the task as being a unit work and the concept of generic Problem Solving Methods used to represent algorithms which could be executed and 'instrumented' (examined for CPU, memory and disk usage) for resource usage sees a move towards a utility market where cloud users could map their requirements to tasks. Again, the issues around generating assertions for framework concepts and relationships are even greater than for CSA. To create tasks and PSM for even a constrained problem domain would require much effort.

The pricing case study built on the requirements case study to successfully demonstrate the elements of the proposed framework could be interconnected. It is possible to obtain pricing information from cloud service providers as spreadsheets or other raw data formats. There is a move to provide information as RDF or other semantic mark-ups such as JSON-LD. This makes pricing the most promising area for automation of assertion generation.

The Benefits Management case study was the most comprehensive case study and built on several years' work from a number of researchers. The enhancement of original Benefits Management technique was highly successful and it was possible to synthesise a number of generic benefits from cloud computing investments.

In conclusion, there are a number of promising concepts from the proposed framework and it was possible to show some linkages between the four elements of the proposed framework. The major issue is the amount of effort required to generate enough assertions to prove the value of the proposed framework.

## **9.5 Critical Analysis**

This section will consider the strengths and weaknesses of the semantic framework in terms of five dimensions, which are listed below:

- The general effectiveness of the semantic framework in modelling cloud services
- Applicability and usefulness of the approach in considering Cloud Service Agreements (CSA)
- Applicability and usefulness of the approach in specifying cloud service requirements
- A critique of the approach in considering cloud service pricing
- A critical analysis of the semantic approach to benefits management of cloud computing investments

### **9.5.1 General Effectiveness of the Semantic Framework**

The framework models four aspects of cloud service description, which allow three phases of cloud service description to be specified, Cloud Service Agreements (CSA), cloud requirements and cloud pricing. A monitoring or

oversight aspect of the framework is the benefits managements of cloud computing investments which is pervasive within the framework.

The first major of benefit of the framework is the usage of semantics throughout the framework, which provides greater expressiveness than frameworks that are syntactical or mathematical in nature. This permits sophisticated modelling of concepts and relationships between concepts as terminology, reasoning within the terminology and the creation of new knowledge. The increasing complexity and size of public clouds see expansion of cloud service combinations that can be selected. It would be difficult, if not impossible to describe all conceivable combinations of service compositions to make optimal service composition selections in real time.

The second major benefit of the framework is to take a holistic approach to cloud service description. Many researchers have concentrated on low-level descriptions of infrastructure and Service Level Agreements (SLA). The semantic framework described in this research not only considers low and high level requirements and how they interact with infrastructure, platforms and software as service. The semantic framework also considers a wider range of CSA, requirements, pricing and benefits management. This wider focus of research of cloud service description provides a unique contribution in this area of research.

The third major benefit is the common representation of aspects of the semantic framework. It is possible to map description elements across each of the four areas of research, for example, there are many common concepts and relationships in Customer Agreements (CA), Service Level Agreements (SLA) and Acceptable Use Policies (AUP), for instance contractual description elements. The terminology and assertions generated from semantic framework and case studies described in the thesis can also be mapped to external semantic frameworks, such as legal semantic frameworks and ontologies.

The semantic framework developed in this research requires potential users have to learn a number of semantic modelling techniques and toolsets to use the framework. This can be resolved by usage of online with a high degree of ease of use, for example tools such as WebProtege or the development of dedicated software to guide users through a constrained set of semantic framework elements, using a dedicated user interface.

An area for improvement for the semantic framework developed in this research is to map and integrate the four framework elements of CSA, cloud service requirements, cloud pricing and benefits management for cloud computing investments. Although some discussion has been made on commonality further work is required for full integration.

There are significant overlaps and mappings between the four framework elements have been identified. The ultimate goal of the current research and future work is to create a unified semantic framework for cloud service description. This concern is addressed in the further work section later in the thesis.

### **9.5.2 Cloud Service Agreements (CSA)**

The semantic framework is highly effective at modelling CSA. There was an extensive analysis of the literature in this area through primary and secondary research. The results from the analysis was a comprehensive range of terminology that was created for Customer Agreements, Service Level Agreements and Acceptable use policies. Relationships were established between the three elements, as well as possible linkages to external frameworks.

Researchers and practitioners now have the foundations for development of toolsets to analyse or guide the CSA process.

### **9.5.3 Requirements**

The semantic framework developed shows a clear description of terminology for cloud service description and a process for mapping high level requirements to low level requirements, which in turn can be mapped to provider service descriptions.

The case study provided demonstrates the effectiveness of the approach, showing a progression of high level requirements to service provider mapping. The effectiveness of the approach is confirmed in the fact that the requirements can be captured over the whole hierarchy and the expressiveness of the framework is proven.

### **9.5.4 Pricing**

Pricing forms an integral part of the semantic framework it appears in CSA, requirements, benefits management as well as a unique workflow design developed as part of the semantic framework for cloud service pricing.

Problem Solving Ontology (PSO) was used to define tasks that were able to map generic problem solving methods and domain ontology. Notional usage metrics for CPU, memory and storage were then used to arrive at pricing for cloud service usage.

The advances produced by this pricing approach are listed below:

- Improved integration with CSA, requirements and Benefits Management semantic information
- A clear and precise workflow for pricing cloud services using semantics
- Greater re-use pricing information through generic problem solving methods that have been instrumented for CPU, memory and storage

which have a notional price which can mapped to a number cloud resources and their prices

#### **9.5.5 Benefits Management**

The benefits management aspects of the framework are pervasive. Benefits management 'drivers' occur before CSA are created and can be seen within CSA, requirements and pricing. Benefits management can also be seen as reviewing the outputs from CSA, requirements and benefits management processes.

The semantic framework for benefits management provided a number of key advantages listed below:

- The improvement of an existing “syntactical” technique by adding semantic description
- Creation of an approach for semantically defining and encoding benefits information
- Establishment of an on-line resource as ontology for a number of case studies derived from primary and secondary sources encoded in the semantic framework

The main issues with development of the semantic framework for benefits management is the learning curve for users of the traditional Benefits Management framework. This problem could be solved by development of a graphical user interface which would present users with diagrams seen in the existing Benefits Management approach, to which semantic information would then be added.



### 9.5.6 Conclusions

The development of the semantic framework as a whole presents a number of advantages over existing 'syntactical'/mathematical approaches to cloud service specifications.

Firstly, valuable description information is not lost and common terminology elements between aspects of description can be defined and mapped. There is a seamless integration between elements of the semantic framework, CSA elements can map to requirements, requirements can map to pricing and benefits management can map to all elements. Although, further work is required to completely map all terminology and assertions across the individual framework elements the foundations are in place.

Secondly, the differences in cloud service providers are mapped to common terminology, so that services can be compared at a concept and relationship level. Notional pricing allows users to compare and select services mapped to CSA and requirements, rather than having to work through many combinations of possible services to discover optimal pricing.

Finally, greater reuse is achieved by using a common problem solving foundation for the semantic frameworks by creation of tasks based on generic problem solving methods that are instrumented for notional pricing.

The semantic framework developed requires potential users have an understanding of semantic terminology to integrate and test interconnections between the four framework elements. This can be resolved by generating bespoke tools and applications to support naïve users in entering assertion information into the framework. Further Integration of the four framework elements is required, this is addressed in future work on a unified frame for cloud service description (below).

## **9.6 Comparison to other Frameworks for Cloud Computing Service Description**

Frameworks for describing cloud computing services are characterised by the usage of simplistic 'syntactical' approaches which use mathematical formulas, networks or entity relationship style approaches. The deficiency in the expression in these techniques were the primary motivation in developing the proposed framework based on semantics.

Existing requirements frameworks developed for software engineering do not consider agreements and pricing which are major elements of the cloud computing service description. The concentration on low level Service Level Agreements means much information is lost in many service description models.

Many pricing models found in research concentrate on a small number of variables such as CPU or memory usage which can be modelled mathematically.

Business value from cloud computing investments has only been considered at a superficial level by many researchers. It is important to create comprehensive models for business value as this will be the deciding factor when making successful cloud computing service investments.

## **9.7 Main Contributions and Conclusions**

This section describes the main contributions of each of the framework elements and as a whole. Conclusions will be reached relating the contributions from the research to the criteria for success defined at the start of the thesis and to the research question posed.

### **9.7.1 Cloud Service Agreements**

Organisations such as CSCC, NIST and ENISA have identified the need for a precise and clear description of Cloud Service Agreements (CSA). This research has examined three artifacts used in CSA that is Customer Agreements (CA), Service Level Agreements (SLA) and Acceptable Use Policies (AUP).

The unique contribution of this research is to model aspects of CSA as semantic description framework. This has provided definitions for some aspects of the three CSA artifacts that are clear, precise and unambiguous. The models were developed using a description logic based framework, with content that can be verified and audited by domain experts. This approach is well established in fields such as bioinformatics.

A number of descriptions of the semantic framework have been uploaded to WebProtégé, a web-based platform for ontology development hosted by Stanford University. This allows researchers to view, enhance and download the implemented ontology. These descriptions can be found in the appendices of this thesis.

Service Level Agreements (SLA) cover a number of subject areas such as computer science, business administration, law and ethics. This requires researchers to be proficient in a broad range of subject areas or be part of a larger team of researchers. General research into this area found by literature review, has focused on narrow areas, for example computer science has concentrated on the implementation of SLA. There is very little research into Customer Agreements and Acceptable Use Policy, beyond legally focused research. One of the advantages of semantic description developed and described in thesis is that the work of a number of researchers work can be merged and be mapped against elements of CSA. There are a number of overlapping areas within the three artifacts such as

pricing and legal aspects that can be mapped to provide common concepts, relations and assertions.

Description Logic based semantic definition allows clear precise and unambiguous representations of CSA models, this models the rich semantics involved in CSA. The semantic description can be used to produce representations that are amenable to software processing and, can therefore support computerised decision support systems to aid users of CSA to verify their rights and obligations provided by the agreements.

### **9.7.2 Requirements**

Future work will see the implementation being expanded to allow for a simpler specification of knowledge components such as tasks, domain knowledge, problem-solving methods and bridges. In future case studies, more complex brokerage will be used. Security will be included in the future version of the requirements ontology as it is a major emerging area in cloud computing.

### **9.7.3 Pricing**

A problem-solving semantic description driven approach to pricing cloud computing services has been described. A pricing semantic description framework was overlaid on semantic description of requirements and general problem solving definitions.

A workflow that utilised the semantic description was defined that used the problem solving semantic description, which mapped tasks from the semantic description onto generic problem solving methods and domain specific semantic description for a knowledge domain

A closed feedback loop was used to feed information generated by cloud APIs back into the semantic description so that requirements could be

adjusted and cloud instance choice changed in the light of new cloud performance metrics.

The model is highly suited to computer-oriented cloud computing requirements. The problem solving semantic description framework provides a set of off-the-shelf concepts and semantic description components that have been designed and refined across a number of frameworks. The pricing process can be seen as a problem-solving exercise and thus fits into the problem-solving semantic description. The semantic description of mediation effectively maps user-requirements to pricing. Collaborative development and mapping strategies are simplified by usage of a common problem solving semantic description.

User requirements are also overlaid over problem-solving semantic description. This provides a shorter framework development time, many other frameworks are developed from scratch, making development more costly and time consuming.

#### **9.7.4 Advantages of a Semantic Description Framework for Benefits Management**

There are a number of advantages in developing a semantic framework for benefits management cloud computing investments. The usage of semantic modelling techniques improves the expressive quality of the techniques and tools found within the original Benefits Management approach. An example could be the Benefits Dependency Network (BDN) which has linkages between enablers, change and benefits that are more expressive than using a simple (plain) network pattern.

Description logics allow reasoning to take place across the semantic description. This has been demonstrated using the SPARQL language. An example can be seen in the BDN where “Cloud Computing Enablers that create change for the financial benefits for strategic investments” can be

found. The reasoning mechanism is more powerful and flexible than that found in technologies such as relation databases and, the terminology, relationships and assertions can be changed in the light of new knowledge more easily. Knowledge can be 'created' by concepts such as multiple-inheritance of knowledge derived through reasoning.

The use of collaborative semantic description tools such as WebProtégé are ideally suited to benefits management development. Stakeholders collaborate to define and edit terminology and assertions. The collaborative tools provide change notification and auditing required in a multi-stakeholder environment.

There has been heavy investment in cloud computing, which is set to increase over the next decade. It is important to consider the benefits cloud computing will bring to organisations. This research has laid the foundations for considering what the likely benefits from cloud computing investments are and, has structured them into an appropriate knowledge representation framework and developed description based ontology to provide proof of concept.

### **9.7.5 Conclusions**

The contributions to knowledge can be summarised in the following list:

- Greater expressivity from the usage of semantic approaches when compared to current 'syntactical'/Mathematical techniques
- A broader framework considering high and low level requirements and a wider range of CSA and benefits management elements
- The ability to map terminology (concepts and relationships) across the four semantic framework elements and external semantic frameworks

- The capability to abstract terminology from cloud service provider descriptions
- Provide a common method of representation using a problem solving approach, with reuse of problem solving methods that have been instrumented for pricing

The criteria for success will now be addressed from section 1.6, as a selection of an appropriate semantic representation amenable to machine representation, application to cloud service scenarios and development of working semantic models.

The literature review identified a number of appropriate semantic representations which were considered in detail. It was found a description logic based approach, with a specialisation of problem solving semantics, provided the most appropriate approach, this delivered a formality appropriate to description of cloud service description that was amenable to machine representation.

A number of case studies were encoded using the framework across the four framework elements. The encodings were presented as ontology on collaborative platforms. This presents positive evidence to their applicability to cloud service scenarios.

Semantic models were developed at a theoretical level and implemented at a research level that would be familiar to investigators and practitioners in this field of study. Further work would be required to allow naïve users to fully utilise the models.

The research question is “Can Cloud Service Agreements, requirements for cloud services, pricing of cloud services and benefits derived from cloud services be modelled using semantic techniques, to aid customer self-service of cloud resources?”

The four elements identified can be modelled using semantic techniques delivering a unique contribution to research into cloud computing service description. The framework will guide, describe and audit a cloud service consumer taking self-service service decisions at a theoretical level and further work would support the naïve user in this process.

## **9.8 Future Work**

Future work will see an expansion of the four framework elements and generate a greater number of case studies on public semantic framework and ontology platforms. Development of toolsets and software systems that can be used by naïve users will increase the uptake of the semantic framework.

The integration of the four framework elements and integration with semantic frameworks will see a unified framework for cloud service description.

### **9.8.1 Expansion of Semantic Framework via the Web Protégé Ontology**

The terminology of the semantic framework can be improved by an internal review and by peer review of the WebProtégé projects, which is designed to provide a collaborative approach to semantic description development.

A number of case studies have been analysed, however, further work is underway to add additional assertions to the semantic description through the analysis of further case studies.

### **9.8.2 Usage of Semantic Framework by Organisations**

WebProtégé is designed for domain experts and non-technical knowledge engineers. Further work will involve the definition of input forms for the entry benefits information. A number of client interfaces are being developed to provide rich user interfaces for non-expert users for the TBox.



The TBox and RBox for the ontology described in this research can be downloaded from WebProtégé [205] and the ABox is available on request. Organisations can use the TBox/RBox to develop their own benefits managements ontology (by defining an ABox) using WebProtégé or a custom user interface (which is under development).

### **9.8.3 Cloud Service Agreements**

The main areas of future work are to expand CA and AUP, as there is relatively little research on these areas in comparison to research on SLA. Many researchers have considered the implementation of SLA when operating in interconnected cloud services, which can be fed into the specifications and requirements of SLA.

A number semantic description model implementations for CSA have been provided. The portfolio of models needs to be expanded to include all aspects of the three artifacts described in CSCC [6], as Individual ontologies need to be mapped and audited to produce a unified semantic description for CSA. Software to support semantic description generation and maintenance for domain experts is required.

There are emerging standards in cloud computing, such as those promulgated by the Open Cloud Computing Interface Working Group (OCCI-WG) [208]. It is important to relate any future work to emerging standards in areas such as pricing and service agreements.

Future work will see development of a number of cloud services (such as a financial calculation service), with semantic description of service agreements. These services will implement Customer Agreements, Acceptable Use Policies and Service Level Agreement descriptions which describe and complement the provision of the service.

#### **9.8.4 Semantic Description for Requirements**

Future work will see the implementation for semantic description of requirements being expanded to allow for a simpler specification of knowledge components such as tasks, domain knowledge, problem-solving methods and bridges. In future case studies, more complex brokerage will be used. Security will be included in the future version of the requirements ontology as it is a major emerging topic in cloud computing.

#### **9.8.5 Pricing**

Work is underway to develop semantic description for two additional problem domains. Firstly, a quantitative analysis service for financial services using the QuantLib [209] financial calculation library. Secondly, a DNA probe design [210] service that can be used to identify medical conditions. The rationale for development of these two domains is that they both require large amounts of processing resources and benefit from lower-cost and faster-processing times offered by cloud computing. Code libraries are being 'instrumented' to obtain metrics for CPU, memory and storage requirements. This information will be uploaded to the WebProtégé semantic description platform.

#### **9.8.6 Benefits Management**

A number of case studies have been analysed and encoded into the semantic framework, however, further work is underway to add additional assertions to the semantic framework through the analysis of further case studies.

Toolsets to allow easier benefits management case study encoding are required. Matching is a key approach which can be used to match enablers, to changes and the benefits generated from stakeholder groups. Mao et al.

[211] describe a mapping approach based on Vector Space Models (VSM), which allow textual descriptions of ontology elements to be mapped.

#### **9.8.7 Towards a Unified Semantic Framework for Cloud Service Description**

An enhancement of the framework is seen in unifying the elements of the current framework. This will involve the greater integration of the four framework elements (Cloud Service Agreements (CSA), requirements, pricing and benefits management for cloud computing investment) and their sub-elements. Terminology (concepts and relationships) will be mapped to produce a unified semantic framework that will be implemented as a problem solving description logic semantic framework.

This will deliver a complete workflow for cloud service description from initial benefits management analysis, through CSA, requirements and pricing to final benefits management analysis.

A further literature review will be carried out to consider semantic frameworks developed by other researchers, for example law and security, to encode and map their work into the unified framework, an obvious starting point are legal and contractual frameworks.

## References

- [1] “EC2 Instance Pricing – Amazon Web Services (AWS).” [Online]. Available: <https://aws.amazon.com/ec2/pricing/>. [Accessed: 09-Mar-2016].
- [2] S. Hendrickson, S. Sturdevant, T. Harter, V. Venkataramani, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau, “Serverless computation with openlambda,” *Elastic*, vol. 60, p. 80, 2016.
- [3] I. Brandic, D. Music, P. Leitner, and S. Dustdar, “Vieslaf framework: Enabling adaptive and versatile sla-management,” in *Grid Economics and Business Models*, Springer, 2009, pp. 60–73.
- [4] J. Ward and E. Daniel, *Benefits Management: How to Increase the Business Value of Your IT Projects*. Wiley, 2012.
- [5] S. Bradshaw, C. Millard, and I. Walden, “Contracts for clouds: comparison and analysis of the Terms and Conditions of cloud computing services,” *International Journal of Law and Information Technology*, vol. 19, no. 3, pp. 187–223, 2011.
- [6] Cloud Standards customer Council, “Public Cloud Service Agreements: What to Expect and What to Negotiate.” 2013.
- [7] H. Cooper, *Research synthesis and meta-analysis: A step-by-step approach*, vol. 2. Sage publications, 2015.
- [8] A. O’hear, “An introduction to the philosophy of science,” 1993.
- [9] M. Armbrust *et al.*, “A view of cloud computing,” *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.
- [10] R. L. Grossman, “The Case for Cloud Computing,” *IT Professional*, vol. 11, no. 2, pp. 23–27, Apr. 2009.
- [11] A. Shkapsky, M. Yang, M. Interlandi, H. Chiu, T. Condie, and C. Zaniolo, “Big data analytics with datalog queries on spark,” presented at the Proceedings of the 2016 International Conference on Management of Data, 2016, pp. 1135–1149.
- [12] M. Stonebraker, S. Madden, and P. Dubey, “Intel big data science and technology center vision and execution plan,” *ACM SIGMOD Record*, vol. 42, no. 1, pp. 44–49, 2013.
- [13] P. Mell and T. Grance, “The NIST definition of cloud computing,” 2011.

- [14] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.
- [15] A. K. Sharma and P. Soni, "Mobile cloud computing (MCC): open research issues," *International Journal of Innovations in Engineering and Technology (IJJET)*, vol. 2, no. 1, pp. 24–27, 2013.
- [16] S. K. Datta, C. Bonnet, and J. Haerri, "Fog Computing architecture to enable consumer centric Internet of Things services," presented at the Consumer Electronics (ISCE), 2015 IEEE International Symposium on, 2015, pp. 1–2.
- [17] E. Jonas, S. Venkataraman, I. Stoica, and B. Recht, "Occupy the Cloud: Distributed computing for the 99%," *arXiv preprint arXiv:1702.04024*, 2017.
- [18] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.
- [19] M. Hogan, F. Liu, A. Sokol, and J. Tong, "Nist cloud computing standards roadmap," *NIST Special Publication*, vol. 35, 2011.
- [20] A. Sheth and A. Ranabahu, "Semantic Modeling for Cloud Computing, Part 1," *Internet Computing, IEEE*, vol. 14, no. 3, pp. 81–83, Jun. 2010.
- [21] C. Ward, M. J. Boco, R. N. Chang, and L. Z. Luan, "A generic SLA semantic model for the execution management of e-business outsourcing contracts," in *E-Commerce and Web Technologies*, Springer, 2002, pp. 363–376.
- [22] R. Greenwell, X. Liu, and K. Chalmers, "Benefits Management of Cloud Computing Investments," *International Journal of Advanced Computer Science and Applications(IJACSA)*, vol. 5, no. 7, 2014.
- [23] I. Arpaci, "Understanding and predicting students' intention to use mobile cloud storage services," *Computers in Human Behavior*, vol. 58, pp. 150–157, 2016.
- [24] S. Butt, H. A. Lagar-Cavilla, A. Srivastava, and V. Ganapathy, "Self-service cloud computing," presented at the Proceedings of the 2012 ACM conference on Computer and communications security, 2012, pp. 253–264.

- [25] A. Menychtas *et al.*, “4CaaS marketplace: An advanced business environment for trading cloud services,” *Future Generation Computer Systems*, 2014.
- [26] Z. Cao, J. Lin, C. Wan, Y. Song, Y. Zhang, and X. Wang, “Optimal Cloud Computing Resource Allocation for Demand Side Management in Smart Grid,” *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1943–1955, 2017.
- [27] H. Zhang, H. Jiang, B. Li, F. Liu, A. V. Vasilakos, and J. Liu, “A framework for truthful online auctions in cloud computing with heterogeneous user demands,” *IEEE Transactions on Computers*, vol. 65, no. 3, pp. 805–818, 2016.
- [28] E. S. Dove *et al.*, “Genomic cloud computing: legal and ethical points to consider,” *European Journal of Human Genetics*, vol. 23, no. 10, p. 1271, 2015.
- [29] Z. Xiao and Y. Xiao, “Security and privacy in cloud computing,” *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 843–859, 2013.
- [30] S. Pearson, “Privacy, security and trust in cloud computing,” in *Privacy and Security for Cloud Computing*, Springer, 2013, pp. 3–42.
- [31] B. Sotomayor, R. S. Montero, I. M. Llorente, and I. Foster, “Virtual infrastructure management in private and hybrid clouds,” *Internet Computing, IEEE*, vol. 13, no. 5, pp. 14–22, 2009.
- [32] N. Grozev and R. Buyya, “Inter-Cloud architectures and application brokering: taxonomy and survey,” *Software: Practice and Experience*, 2012.
- [33] A. Abdelmaboud, D. N. Jawawi, I. Ghani, A. Elsafi, and B. Kitchenham, “Quality of service approaches in cloud computing: A systematic mapping study,” *Journal of Systems and Software*, vol. 101, pp. 159–179, 2015.
- [34] D. Nurmi *et al.*, “The eucalyptus open-source cloud-computing system,” presented at the Cluster Computing and the Grid, 2009. CCGRID’09. 9th IEEE/ACM International Symposium on, 2009, pp. 124–131.
- [35] R. N. Calheiros, C. Vecchiola, D. Karunamoorthy, and R. Buyya, “The Aneka platform and QoS-driven resource provisioning for elastic applications on hybrid Clouds,” *Future Generation Computer Systems*, vol. 28, no. 6, pp. 861–870, 2012.

- [36] Z. Mao, Y. Shang, C. Liu, and I. Chen, "Utility-based Price Proportion in Cloud Resource Allocation.," *Information Technology Journal*, vol. 12, no. 22, pp. 6882–6886, 2013.
- [37] R. Greenwell, X. Liu, and K. Chalmers, "Pricing Intelligence as a service for Cloud computing.," *IEEE Cloudcom 2013*, 2013.
- [38] F. R. Carlson, "Security Analysis of Cloud Computing," *arXiv preprint arXiv:1404.6849*, 2014.
- [39] N. Gonzalez *et al.*, "A quantitative analysis of current security concerns and solutions for cloud computing," *Journal of Cloud Computing*, vol. 1, no. 1, pp. 1–18, 2012.
- [40] R. Buyya, R. Ranjan, and R. N. Calheiros, "Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services," in *Algorithms and architectures for parallel processing*, Springer, 2010, pp. 13–31.
- [41] C. Rong, S. T. Nguyen, and M. G. Jaatun, "Beyond lightning: A survey on security challenges in cloud computing," *Computers & Electrical Engineering*, vol. 39, no. 1, pp. 47–54, 2013.
- [42] K. Grolinger, W. A. Higashino, A. Tiwari, and M. A. Capretz, "Data management in cloud environments: NoSQL and NewSQL data stores," *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 2, no. 1, p. 22, 2013.
- [43] A. N. Toosi, R. N. Calheiros, and R. Buyya, "Interconnected Cloud Computing Environments: Challenges, Taxonomy, and Survey," *ACM Computing Surveys (CSUR)*, vol. 47, no. 1, p. 7, 2014.
- [44] S. Wind and H. Schrödl, "Requirements engineering for cloud computing: a comparison framework," in *Proceedings of the International conference on Web Information Systems Engineering*, Berlin, Heidelberg, 2011, pp. 404–415.
- [45] D. Fensel *et al.*, "The Unified Problem-solving Method Development Language UPML," *Knowledge and Information Systems, Springer-Verlag London Limited*, vol. 5, no. 1, pp. 83–131, 2003.
- [46] H. J. Happel and S. Seedorf, "Applications of ontologies in software engineering," presented at the Proc. of Workshop on Semantic Web Enabled Software Engineering"(SWESE) on the ISWC, 2006, pp. 5–9.
- [47] S. Farfeleder, T. Moser, A. Krall, T. Stalhane, H. Zojer, and C. Panis, "DODT: Increasing requirements formalism using domain ontologies for

- improved embedded systems development, proceedings of the 14th IEEE International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS),” 2011, pp. 271–274.
- [48] I. J. Jureta, J. Mylopoulos, and S. Faulkner, “Revisiting the core ontology and problem in requirements engineering,” presented at the Proceedings of the 16th IEEE International Conference on Requirements Engineering (RE’08), 2008, 2008, pp. 71–80.
  - [49] P. Bogg, G. Beydoun, and G. Low, “Problem-solving methods in agent-oriented software engineering,” *Information Systems Development, Springer New York*, vol. 4, no. 1, pp. 243–254, 2011.
  - [50] T. Kiemes, D. Oberle, and F. Novelli, “Towards a reusable and executable pricing model in the internet of services,” presented at the Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services, 2010, pp. 722–729.
  - [51] D.-W.-I. S. Lehmann and P. Buxmann, “Pricing strategies of software vendors,” *Business & Information Systems Engineering*, vol. 1, no. 6, pp. 452–462, 2009.
  - [52] T. T. Nagle, J. E. Hogan, and J. Zale, *The Strategy and Tactics of Pricing: A Guide to Growing More Profitably*. Prentice Hall PTR, 2010.
  - [53] V. Abhishek, I. A. Kash, and P. Key, “Fixed and market pricing for cloud services,” presented at the Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on, 2012, pp. 157–162.
  - [54] I. A. Kash and P. B. Key, “Pricing the Cloud,” *IEEE Internet Computing*, vol. 20, no. 1, pp. 36–43, 2016.
  - [55] G. Di Modica and O. Tomarchio, “A Semantic Model to Characterize Pricing and Negotiation Schemes of Cloud Resources,” presented at the CLOUD COMPUTING 2012, The Third International Conference on Cloud Computing, GRIDs, and Virtualization, 2012, pp. 18–23.
  - [56] B. Wagner and A. Sood, “Economics of Resilient Cloud Services,” presented at the Software Quality, Reliability and Security Companion (QRS-C), 2016 IEEE International Conference on, 2016, pp. 368–374.
  - [57] M. Villamizar *et al.*, “Cost comparison of running web applications in the cloud using monolithic, microservice, and AWS Lambda architectures,” *Service Oriented Computing and Applications*, vol. 11, no. 2, pp. 233–247, 2017.



- [58] J. Weinman, *Clouconomics: The business value of cloud computing*. John Wiley & Sons, 2012.
- [59] P. Mell and T. Grance, "The NIST definition of cloud computing (draft)," *NIST special publication*, vol. 800, no. 145, p. 7, 2011.
- [60] J. Peppard, J. Ward, and E. Daniel, "Managing the realization of business benefits from IT investments," *MIS Quarterly Executive*, vol. 6, no. 1, pp. 1–11, 2007.
- [61] T. Dillon, C. Wu, and E. Chang, "Cloud computing: issues and challenges," presented at the Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on, 2010, pp. 27–33.
- [62] M. Uschold and M. Gruninger, "Ontologies: Principles, methods and applications," *The knowledge engineering review*, vol. 11, no. 2, pp. 93–136, 1996.
- [63] B. Chandrasekaran, J. R. Josephson, and V. R. Benjamins, "What are ontologies, and why do we need them?," *IEEE Intelligent Systems and their applications*, vol. 14, no. 1, pp. 20–26, 1999.
- [64] B. Di Martino, G. Cretella, and A. Esposito, "Semantic and agnostic representation of cloud patterns for cloud interoperability and portability," presented at the Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on, 2013, vol. 2, pp. 182–187.
- [65] A. Jula, E. Sundararajan, and Z. Othman, "Cloud computing service composition: A systematic literature review," *Expert Systems with Applications*, vol. 41, no. 8, pp. 3809–3824, 2014.
- [66] F. Rosenberg, P. Celikovic, A. Michlmayr, P. Leitner, and S. Dustdar, "An end-to-end approach for QoS-aware service composition," presented at the Enterprise Distributed Object Computing Conference, 2009. EDOC'09. IEEE International, 2009, pp. 151–160.
- [67] G. Zou, Y. Chen, Y. Yang, R. Huang, and Y. Xu, "AI planning and combinatorial optimization for web service composition in cloud computing," presented at the Proc international conference on cloud computing and virtualization, 2010, pp. 1–8.
- [68] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in cloud," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 42–57, 2013.

- [69] H. Bao and W. Dou, "A QoS-aware service selection method for cloud service composition," presented at the Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th International, 2012, pp. 2254–2261.
- [70] D. de Oliveira, E. Ogasawara, F. Baião, and M. Mattoso, "Scicumulus: A lightweight cloud middleware to explore many task computing paradigm in scientific workflows," presented at the Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on, 2010, pp. 378–385.
- [71] S. Sundareswaran, A. Squicciarini, and D. Lin, "A brokerage-based approach for cloud service selection," presented at the Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on, 2012, pp. 558–565.
- [72] D. Comer, "Ubiquitous B-tree," *ACM Computing Surveys (CSUR)*, vol. 11, no. 2, pp. 121–137, 1979.
- [73] P. Patel, A. Ranabahu, and A. Sheth, "Service Level Agreement in cloud computing," presented at the Cloud Workshops at OOPSLA, 2009.
- [74] T. V. Pham, H. Jamjoom, K. Jordan, and Z.-Y. Shae, "A service composition framework for market-oriented high performance computing cloud," presented at the Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing, 2010, pp. 284–287.
- [75] S. Staab, S. Santini, F. Nack, L. Steels, and A. Maedche, "Emergent semantics," *IEEE Intelligent Systems*, vol. 17, no. 1, pp. 78–86, 2002.
- [76] T. Landauer, P. Foltz, and D. Laham, "Introduction to Latent Semantic Analysis," *Discourse Processes*, 1998.
- [77] C. Zeng, X. Guo, W. Ou, and D. Han, "Cloud Computing Service Composition and Search Based on Semantic," in *Cloud Computing*, vol. 5931, M. Jaatun, G. Zhao, and C. Rong, Eds. Springer Berlin / Heidelberg, 2009, pp. 290–300.
- [78] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2010.
- [79] A. Dennis, B. H. Wixom, and D. Tegarden, *Systems analysis and design: An object-oriented approach with UML*. John Wiley & Sons, 2015.

- [80] T. Pandey and S. Srivastava, "Comparative analysis of formal specification languages Z, VDM, B," *International Journal of Current Engineering and Technology E-ISSN*, pp. 2277–4106, 2015.
- [81] F. Baader, I. Horrocks, and U. Sattler, "Description logics," *Foundations of Artificial Intelligence*, vol. 3, pp. 135–179, 2008.
- [82] M. Krötzsch, F. Simancik, and I. Horrocks, "Description Logics.," *IEEE Intelligent Systems*, vol. 29, no. 1, pp. 12–19, 2014.
- [83] D. Allemang and J. Hendler, *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*. Elsevier Science, 2011.
- [84] B. C. Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, and U. Sattler, "OWL 2: The next step for OWL," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 6, no. 4, pp. 309–322, 2008.
- [85] B. N. Grosz, I. Horrocks, R. Volz, and S. Decker, "Description logic programs: combining logic programs with description logic," presented at the Proceedings of the 12th international conference on World Wide Web, 2003, pp. 48–57.
- [86] I. Horrocks, "What Are Ontologies Good For?," in *Evolution of Semantic Systems*, Springer, 2013, pp. 175–188.
- [87] I. Horrocks, "Applications of description logics: State of the art and research challenges," in *Conceptual Structures: Common Semantics for Sharing Knowledge*, Springer, 2005, pp. 78–90.
- [88] F. Baader and U. Sattler, "Tableau algorithms for description logics," presented at the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods, 2000, pp. 1–18.
- [89] V. Haarslev, R. Möller, and A.-Y. Turhan, "Exploiting pseudo models for TBox and ABox reasoning in expressive description logics," presented at the IJCAR, 2001, vol. 1, pp. 61–75.
- [90] B. Motik, R. Shearer, and I. Horrocks, "Hypertableau reasoning for description logics," *Journal of Artificial Intelligence Research*, vol. 36, no. 1, pp. 165–228, 2009.
- [91] Protege, "The Protégé Ontology Editor and Knowledge Acquisition System," 2014. [Online]. Available: <http://protege.stanford.edu/>. [Accessed: 24-Nov-2014].

- [92] B. Parsia, N. Matentzoglou, R. S. Gonçalves, B. Glimm, and A. Steigmiller, "The OWL Reasoner Evaluation (ORE) 2015 Resources," presented at the International Semantic Web Conference, 2016, pp. 159–167.
- [93] B. Glimm, I. Horrocks, B. Motik, G. Stoilos, and Z. Wang, "HermiT: an OWL 2 reasoner," *Journal of Automated Reasoning*, vol. 53, no. 3, pp. 245–269, 2014.
- [94] D. Tsarkov and I. Horrocks, "FaCT++ description logic reasoner: System description," *Automated reasoning*, pp. 292–297, 2006.
- [95] B. Parsia and E. Sirin, "Pellet: An owl dl reasoner," presented at the Third international semantic web conference-poster, 2004, vol. 18, p. 13.
- [96] F. Baader, I. Horrocks, C. Lutz, and U. Sattler, *An Introduction to Description Logic*. Cambridge University Press, 2017.
- [97] O. Bodenreider, B. Smith, A. Kumar, and A. Burgun, "Investigating subsumption in SNOMED CT: An exploration into large description logic-based biomedical terminologies," *Artificial intelligence in medicine*, vol. 39, no. 3, pp. 183–195, 2007.
- [98] P. Tetlow, J. Z. Pan, D. Oberle, E. Wallace, M. Uschold, and E. Kendall, "Ontology driven architectures and potential uses of the semantic web in systems and software engineering," *W3C Working Draft*, 2005.
- [99] M. Crubézy and M. A. Musen, "Ontologies in Support of Problem Solving," in *Handbook on Ontologies*, 2003, pp. 321–342.
- [100] P. Zave and M. Jackson, "Four dark corners of requirements engineering," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 6, no. 1, pp. 1–30, 1997.
- [101] A. Classen, "Problem-Oriented Feature Interaction Detection in Software Product Lines," 2007.
- [102] V. Castaneda, L. Ballejos, M. L. Caliusco, and M. R. Galli, "The Use of Ontologies in Requirements Engineering," *Global Journal of Research Engineering*, vol. 10, no. 6, 2010.
- [103] L. Youseff, M. Butrico, and D. D. Silva, "Towards a unified ontology of cloud computing," in *Proc. of the Grid Computing Environments Workshop (GCE08)*, Austin, Texas, 2008.

- [104] F. Amato and F. Moscato, "Exploiting cloud and workflow patterns for the analysis of composite cloud services," *Future Generation Computer Systems*, vol. 67, pp. 255–265, 2017.
- [105] F. Amato and F. Moscato, "Pattern-based orchestration and automatic verification of composite cloud services," *Computers & Electrical Engineering*, vol. 56, pp. 842–853, 2016.
- [106] D. Fang, X. Liu, I. Romdhani, P. Jamshidi, and C. Pahl, "An agility-oriented and fuzziness-embedded semantic model for collaborative cloud service search, retrieval and recommendation," *Future Generation Computer Systems*, vol. 56, pp. 11–26, 2016.
- [107] D. Fang, X. Liu, I. Romdhani, and C. Pahl, "An approach to unified cloud service access, manipulation and dynamic orchestration via semantic cloud service operation specification framework," *Journal of Cloud Computing*, vol. 4, no. 1, pp. 1–20, 2015.
- [108] S. Ghazouani and Y. Slimani, "A survey on cloud service description," *Journal of Network and Computer Applications*, 2017.
- [109] D. Oberle, A. Barros, U. Kylau, and S. Heinzl, "A unified description language for human to automated services," *Information Systems*, vol. 38, no. 1, pp. 155–181, 2013.
- [110] A. V. Dastjerdi and R. Buyya, "An Autonomous Time-Dependent SLA Negotiation Strategy for Cloud Computing," *The Computer Journal*, p. bxv053, 2015.
- [111] A. V. Dastjerdi, S. G. H. Tabatabaei, and R. Buyya, "A dependency-aware ontology-based approach for deploying service level agreement monitoring services in Cloud," *Software: Practice and Experience*, vol. 42, no. 4, pp. 501–518, 2012.
- [112] K. P. Joshi, Y. Yesha, and T. Finin, "Automating cloud services life cycle through semantic technologies," *IEEE Transactions on Services Computing*, vol. 7, no. 1, pp. 109–122, 2014.
- [113] A. Immonen, E. Ovaska, J. Kalaoja, and D. Pakkala, "A service requirements engineering method for a digital service ecosystem," *Service Oriented Computing and Applications*, vol. 10, no. 2, pp. 151–172, 2016.
- [114] H. Takabi, J. B. Joshi, and G.-J. Ahn, "Security and privacy challenges in cloud computing environments," *IEEE Security & Privacy*, vol. 8, no. 6, pp. 24–31, 2010.

- [115] A. Souag, R. Mazo, C. Salinesi, and I. Comyn-Wattiau, "Reusable knowledge in security requirements engineering: a systematic mapping study," *Requirements Engineering*, vol. 21, no. 2, pp. 251–283, 2016.
- [116] W.-T. Tsai and Q. Shao, "Role-based access-control using reference ontology in clouds," presented at the Autonomous Decentralized Systems (ISADS), 2011 10th International Symposium on, 2011, pp. 121–128.
- [117] M. Musen, "Building Systems with Ontologies and Problem-Solving Methods." Stanford University, 2001.
- [118] R. Mizoguchi, J. Vanwelkenhuysen, and M. Ikeda, "Task ontology for reuse of problem solving knowledge," *Towards Very Large Knowledge Bases: Knowledge Building & Knowledge Sharing*, vol. 46, p. 59, 1995.
- [119] A. Gómez-Pérez and R. Benjamins, "Overview of knowledge sharing and reuse components: Ontologies and problem-solving methods," 1999.
- [120] W. Yan, C. Zanni-Merk, D. Cavallucci, and P. Collet, "An ontology-based approach for inventive problem solving," *Engineering Applications of Artificial Intelligence*, vol. 27, pp. 175–190, 2014.
- [121] G. Altshuller, *40 Principles: TRIZ keys to innovation*, vol. 1. Technical Innovation Center, Inc., 2002.
- [122] F. Scharffe, O. Zamazal, and D. Fensel, "Ontology alignment design patterns," *Knowledge and information systems*, vol. 40, no. 1, pp. 1–28, 2014.
- [123] M. N. Kamel Boulos and S. Wheeler, "The emerging Web 2.0 social software: an enabling suite of sociable technologies in health and health care education1," *Health Information & Libraries Journal*, vol. 24, no. 1, pp. 2–23, 2007.
- [124] J. S. Brown and R. P. Adler, "Minds on Fire: Open Education, the Long Tail, and Learning 2.0," *EDUCAUSE Review*, vol. 43, no. 1, 2008.
- [125] D. Churchill, "Educational applications of Web 2.0: Using blogs to support teaching and learning," *British Journal of Educational Technology*, vol. 40, no. 1, pp. 179–183, 2009.
- [126] E. Constantinides and S. J. Fountain, "Web 2.0: Conceptual foundations and marketing issues," *Journal of Direct, Data and Digital Marketing Practice*, vol. 9, no. 3, pp. 231–244, 2008.

- [127] J. M. Silva, A. S. M. M. Rahman, and A. E. Saddik, "Web 3.0: a vision for bridging the gap between real and virtual," *International Multimedia Conference*, p. 5, 2008.
- [128] O. Lassila and J. Hendler, "Embracing 'Web 3.0,'" *Internet Computing, IEEE*, vol. 11, no. 3, pp. 90–93, Jun. 2007.
- [129] J. Hendler, "Web 3.0 Emerging," *Computer*, vol. 42, no. 1, pp. 111–113, 2009.
- [130] Á. García-Crespo, J. M. Gómez-Berbís, R. Colomo-Palacios, and F. García-Sánchez, "Digital libraries and Web 3.0. The CallimachusDL approach," *Computers in Human Behavior*, vol. 27, no. 4, pp. 1424–1430, 2011.
- [131] K. Nath and R. Iswary, "What Comes after Web 3.0? Web 4.0 and the Future," *Режим доступа: [https://goo. gl/k5OdA6](https://goo.gl/k5OdA6) (дата обращения: 25.07. 2016)*.
- [132] P. F. Patel-Schneider and I. Horrocks, "A comparison of two modelling paradigms in the Semantic Web," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 5, no. 4, pp. 240–250, 2007.
- [133] O. Hartig, "SPARQL for a web of linked data: semantics and computability," in *The Semantic Web: Research and Applications*, Springer, 2012, pp. 8–23.
- [134] M. Kaminski, B. C. Grau, E. V. Kostylev, B. Motik, and I. Horrocks, "Foundations of Declarative Data Analysis Using Limit Datalog Programs," *arXiv preprint arXiv:1705.06927*, 2017.
- [135] I. Hickson, "HTML Microdata Nightly. W3C Editor's Draft, W3C," 2013.
- [136] C. Luo, Z. Zheng, X. Wu, F. Yang, and Y. Zhao, "Automated structural semantic annotation for RESTful services," *International Journal of Web and Grid Services*, vol. 12, no. 1, pp. 26–41, 2016.
- [137] K. Breitman, M. A. Casanova, and W. Truszkowski, *Semantic web: concepts, technologies and applications*. Springer Science & Business Media, 2007.
- [138] W3C, "Resource Description Framework (RDF) Model and Syntax Specification," 2012. [Online]. Available: <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>. [Accessed: 11-Feb-2012].

- [139] J. Z. Pan and I. Horrocks, "Rdfs (fa) and rdf mt: Two semantics for rdfs," presented at the International Semantic Web Conference, 2003, vol. 2870, pp. 30–46.
- [140] B. C. Hitz *et al.*, "SnoVault and encodeD: A novel object-based storage system and applications to ENCODE metadata," *PloS one*, vol. 12, no. 4, p. e0175310, 2017.
- [141] S. J. Chalk, "SciData: a data model and ontology for semantic representation of scientific data," *Journal of cheminformatics*, vol. 8, no. 1, p. 54, 2016.
- [142] J. Urbani, S. Kotoulas, J. Maassen, F. Van Harmelen, and H. Bal, "OWL reasoning with WebPIE: calculating the closure of 100 billion triples," *The Semantic Web: Research and Applications*, pp. 213–227, 2010.
- [143] H. Boley, S. Tabet, and G. Wagner, "Design rationale of RuleML: A markup language for semantic web rules," presented at the Proceedings of the First International Conference on Semantic Web Working, 2001, pp. 381–401.
- [144] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosz, and M. Dean, "SWRL: A semantic web rule language combining OWL and RuleML," *W3C Member submission*, vol. 21, p. 79, 2004.
- [145] M. O'Connor *et al.*, "Supporting rule system interoperability on the semantic web with SWRL," *The Semantic Web–ISWC 2005*, pp. 974–986, 2005.
- [146] J. Domingue *et al.*, "IRS-III: A broker-based approach to semantic Web services," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 6, no. 2, pp. 109–132, 2008.
- [147] WSML, "WSML - Web Service Modeling Language," 2012. [Online]. Available: <http://www.wsmo.org/wsml/>. [Accessed: 21-Feb-2012].
- [148] WSMO, "Web Service Modeling Ontology," 2012. [Online]. Available: <http://www.wsmo.org/>. [Accessed: 21-Feb-2012].
- [149] WSMX, "WSMX - Web Service Execution Environment." [Online]. Available: <http://www.wsmx.org/>. [Accessed: 21-Feb-2012].
- [150] M. Gaspari, E. Motta, and D. Fensel, "Automatic Selection of Problem Solving Libraries Based on Competence Matching," in *Proceedings of the Workshop on Object-Oriented Technology*, 1999, pp. 10–11.



- [151] S. Dietze, A. Gugliotta, J. Domingue, H. Yu, and M. Mrissa, "An automated approach to Semantic Web Services Mediation," *Service Oriented Computing and Applications*, vol. 4, no. 4, pp. 261–275, 2010.
- [152] Q. Z. Sheng, X. Qiao, A. V. Vasilakos, C. Szabo, S. Bourne, and X. Xu, "Web services composition: A decade's overview," *Information Sciences*, vol. 280, pp. 218–238, 2014.
- [153] J. Bruijn, M. Kerrigan, M. Zaremba, and D. Fensel, "Semantic Web Services," *Handbook on Ontologies*, pp. 617–636, 2009.
- [154] M. Hepp, "Goodrelations: An ontology for describing products and services offers on the web," in *Knowledge Engineering: Practice and Patterns*, Springer, 2008, pp. 329–346.
- [155] M. C. Chibucos *et al.*, "Standardized description of scientific evidence using the Evidence Ontology (ECO)," *Database*, vol. 1, p. 11, 2014.
- [156] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," presented at the Proceedings of the first edition of the MCC workshop on Mobile cloud computing, 2012, pp. 13–16.
- [157] J. Zhu, D. S. Chan, M. S. Prabhu, P. Natarajan, H. Hu, and F. Bonomi, "Improving web sites performance using edge servers in fog computing architecture," presented at the Service Oriented System Engineering (SOSE), 2013 IEEE 7th International Symposium on, 2013, pp. 320–323.
- [158] S. Walk, P. Singer, M. Strohmaier, D. Helic, N. Noy, and M. Musen, "Sequential Usage Patterns in Collaborative Ontology-Engineering Projects," *arXiv preprint arXiv:1403.1070*, 2014.
- [159] T. Tudorache, C. Nyulas, N. F. Noy, and M. A. Musen, "WebProtégé: A collaborative ontology editor and knowledge acquisition tool for the web," *Semantic web*, vol. 4, no. 1, pp. 89–99, 2013.
- [160] A. Sebastian, N. F. Noy, T. Tudorache, and M. A. Musen, "A Generic Ontology for Collaborative Ontology-Development Workflows," in *Proceedings of the 16th international conference on Knowledge Engineering: Practice and Patterns*, Berlin, Heidelberg, 2008, pp. 318–328.
- [161] H. Wang, T. Tudorache, D. Dou, N. F. Noy, and M. A. Musen, "Analysis of User Editing Patterns in Ontology Development Projects," presented at the On the Move to Meaningful Internet Systems: OTM 2013 Conferences, 2013, pp. 470–487.

- [162] M. Strohmaier *et al.*, “How ontologies are made: Studying the hidden social dynamics behind collaborative ontology engineering projects,” *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 20, pp. 18–34, 2013.
- [163] C. Manning, P. Raghavan, and Schutze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- [164] E. Alatrish, “Comparison Some of Ontology,” *Journal of Management Information Systems*, vol. 8, no. 2, pp. 018–024, 2013.
- [165] M. Crubézy, “UPML Ontology - The Unified Problem-solving Method Development Language UPML,” 2003. [Online]. Available: <http://webprotege.stanford.edu/#Edit:projectId=c5bc5e07-7139-4e34-9645-209062eca6ad>.
- [166] R. Greenwell, X. Liu, K. Chalmers, and C. Pahl, “A Task Orientated Requirements Ontology for Cloud Computing Services,” presented at the 6th Intl Conf on Cloud Computing and Services Science Closer, 2016.
- [167] Oracle, “Oracle United States Online CSA - cloud-csa-v030917-us-en-3657533.pdf.” [Online]. Available: <http://www.oracle.com/us/corporate/contracts/cloud-csa-v030917-us-en-3657533.pdf>. [Accessed: 05-Sep-2017].
- [168] IBM, “IBM Cloud Service Agreement.” [Online]. Available: [https://www-05.ibm.com/support/operations/files/pdf/csa\\_us.pdf](https://www-05.ibm.com/support/operations/files/pdf/csa_us.pdf). [Accessed: 05-Sep-2017].
- [169] C. Fürber and M. Hepp, “Using SPARQL and SPIN for Data Quality Management on the Semantic Web,” presented at the Business Information Systems, 2010, pp. 35–46.
- [170] M. Mao, Y. Peng, and M. Spring, “An adaptive ontology mapping approach with neural network based constraint satisfaction,” *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 8, no. 1, pp. 14–25, 2010.
- [171] N. Oldham, K. Verma, A. Sheth, and F. Hakimpour, “Semantic WS-agreement partner selection,” presented at the Proceedings of the 15th international conference on World Wide Web, 2006, pp. 697–706.
- [172] S. K. Garg, S. Versteeg, and R. Buyya, “A framework for ranking of cloud computing services,” *Future Generation Computer Systems*, vol. 29, no. 4, pp. 1012–1023, 2013.

- [173] B. Rimal, A. Jukan, D. Katsaros, and Y. Goeleven, "Architectural Requirements for Cloud Computing Systems: An Enterprise Cloud Approach," *Journal of Grid Computing*, vol. 9, no. 1, pp. 3–26, 2011.
- [174] W. N. Robinson, "Monitoring Web service requirements," *Requirements Engineering Conference, 2003. Proceedings. 11th IEEE International*, pp. 65–74, 8.
- [175] Y. Sun, T. Harmer, A. Stewart, and P. Wright, "Mapping application requirements to cloud resources," in *Euro-Par 2011: Parallel Processing Workshops*, 2012, pp. 104–112.
- [176] V. Nebot and R. Berlanga, "Efficient retrieval of ontology fragments using an interval labeling scheme," *Information Sciences*, vol. 179, no. 24, pp. 4151–4173, 2009.
- [177] Y. Kalfoglou, P. R. Smart, D. Braines, and N. Shadbolt, "POAF: Portable ontology aligned fragments," 2008.
- [178] H. S. Packer, N. Gibbins, and N. R. Jennings, "Collaborative learning of ontology fragments by cooperating agents," 2010.
- [179] Y. O. Yazir *et al.*, "Dynamic resource allocation in computing clouds using distributed multiple criteria decision analysis," in *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, 2010, pp. 91–98.
- [180] T. A. Henzinger, A. V. Singh, V. Singh, T. Wies, and D. Zufferey, "FlexPRICE: Flexible Provisioning of Resources in a Cloud Environment," in *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, 2010, pp. 83–90.
- [181] H. K. Bhargava and S. Sundaresan, "Computing as utility: Managing availability, commitment, and pricing through contingent bid auctions," *Journal of Management Information Systems*, vol. 21, no. 2, pp. 201–227, 2004.
- [182] D. Martin, J. Javier, and C. A. Iglesias Fernandez, "An Ontology for Formalising Agreement Patterns in Auction Markets," 2010.
- [183] J. Huang, R. J. Kauffman, and D. Ma, "Pricing strategy for cloud computing: A damaged services perspective," *Decision Support Systems*, vol. 78, pp. 80–92, 2015.
- [184] OCCI-WG, "Open Cloud Computing Interface | Open Standard | Open Community." [Online]. Available: <http://occi-wg.org/>. [Accessed: 29-Oct-2014].

- [185] E.-G. Talbi, "A unified taxonomy of hybrid metaheuristics with mathematical programming, constraint programming and machine learning," in *Hybrid Metaheuristics*, Springer, 2013, pp. 3–76.
- [186] D. Debels, B. De Reyck, R. Leus, and M. Vanhoucke, "A hybrid scatter search/electromagnetism meta-heuristic for project scheduling," *European Journal of Operational Research*, vol. 169, no. 2, pp. 638–653, 2006.
- [187] C. S. Yeo, S. Venugopal, X. Chu, and R. Buyya, "Autonomic metered pricing for a utility computing service," *Future Generation Computer Systems*, vol. 26, no. 8, pp. 1368–1380, 2010.
- [188] B. Martens, M. Walterbusch, and F. Teuteberg, "Costing of Cloud Computing Services: A Total Cost of Ownership Approach," presented at the System Science (HICSS), 2012 45th Hawaii International Conference on, 2012, pp. 1563–1572.
- [189] D. Agrawal, "Towards the End-to-End Design for Big Data Management in the Cloud: Why, How, and When?," presented at the BTW, 2013, pp. 15–16.
- [190] I. T. Foster and R. K. Madduri, "Science as a service: how on-demand computing can accelerate discovery," presented at the Proceedings of the 4th ACM workshop on Scientific cloud computing, 2013, pp. 1–2.
- [191] C. Wang, K. Ren, W. Lou, and J. Li, "Toward publicly auditable secure cloud data storage services," *Network, IEEE*, vol. 24, no. 4, pp. 19–24, 2010.
- [192] A. Khajeh-Hosseini, D. Greenwood, and I. Sommerville, "Cloud migration: A case study of migrating an enterprise it system to iaas," presented at the Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on, 2010, pp. 450–457.
- [193] N. Convery and K. Ferguson-Boucher, "Storing information in the cloud," *Bulletin of Information and Records Management Society, Issue*, pp. 3–5, 2010.
- [194] V. Chang and G. Wills, "A University of Greenwich Case Study of Cloud Computing," *E-Logistics and E-Supply Chain Management: Applications for Evolving Business*, p. 232, 2013.
- [195] M. Shi, "Capturing strategic competences: cloud security as a case study," *Journal of Business Strategy*, vol. 34, no. 3, pp. 41–48, 2013.

- [196] A. Coleman, M. E. Herselman, and M. Coleman, "Improving Computer-mediated Synchronous Communication of Doctors in Rural Communities through Cloud Computing A Case study of Rural Hospitals in South Africa.," *Oriental Anthropologists*, vol. 13, no. 1, 2013.
- [197] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Sensing as a service model for smart cities supported by internet of things," *Transactions on Emerging Telecommunications Technologies*, vol. 25, no. 1, pp. 81–93, 2014.
- [198] G-Cloud, "G-Cloud - Home of the UK Government G-Cloud Programme G-Cloud," 2013. [Online]. Available: <http://gcloud.civilservice.gov.uk/>. [Accessed: 24-Nov-2014].
- [199] J. Bughin, M. Chui, and J. Manyika, "Clouds, big data, and smart assets: Ten tech-enabled business trends to watch," *McKinsey Quarterly*, vol. 56, 2010.
- [200] O. H. Alhazmi and Y. K. Malaiya, "Evaluating disaster recovery plans using the cloud," presented at the Reliability and Maintainability Symposium (RAMS), 2013 Proceedings-Annual, 2013, pp. 1–6.
- [201] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless communications and mobile computing*, vol. 13, no. 18, pp. 1587–1611, 2013.
- [202] D. S. Marcon, R. R. Oliveira, M. C. Neves, L. S. Buriol, L. P. Gaspar, and M. P. Barcellos, "Trust-based Grouping for Cloud Datacenters: improving security in shared infrastructures," presented at the IFIP Networking Conference, 2013, 2013, pp. 1–9.
- [203] Y. Han, "On the clouds: a new way of computing," *Information Technology and Libraries*, vol. 29, no. 2, pp. 87–92, 2013.
- [204] Z. Wu, M. Butkiewicz, D. Perkins, E. Katz-Bassett, and H. V. Madhyastha, "Spanstore: Cost-effective geo-replicated storage spanning multiple cloud services," presented at the Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles, 2013, pp. 292–308.
- [205] R. Greenwell, "Benefits Management Web-protege project," 2014. [Online]. Available: <http://webprotege.stanford.edu/#Edit:projectId=d9a84e6c-9448-4921-b38c-e5e4f0e0ef2c>.

- [206] C. Weinhardt *et al.*, “Cloud computing—a classification, business models, and research directions,” *Business & Information Systems Engineering*, vol. 1, no. 5, pp. 391–399, 2009.
- [207] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, and A. Ghalsasi, “Cloud computing — The business perspective,” *Decision Support Systems*, vol. 51, no. 1, pp. 176–189, 2011.
- [208] T. Metsch and A. Edmonds, “Open Cloud Computing Interface–Infrastructure,” presented at the Standards Track, no. GFD-R in The Open Grid Forum Document Series, Open Cloud Computing Interface (OCCI) Working Group, Muncie (IN), 2010.
- [209] M. Choudhry, D. Joannas, G. Landuyt, R. Pereira, and R. Pienaar, “Term Structure Construction Using QuantLib,” in *Capital Market Instruments*, Springer, 2010, pp. 224–233.
- [210] M. D. Croning, D. G. Fricker, N. H. Komiyama, and S. G. Grant, “Automated design of genomic Southern blot probes,” *BMC genomics*, vol. 11, no. 1, p. 74, 2010.
- [211] M. Mao, J. Li, and M. Humphrey, “Cloud auto-scaling with deadline and budget constraints,” in *Grid Computing (GRID), 2010 11th IEEE/ACM International Conference on*, 2010, pp. 41–48.

## Appendix A - Abbreviations and Acronyms

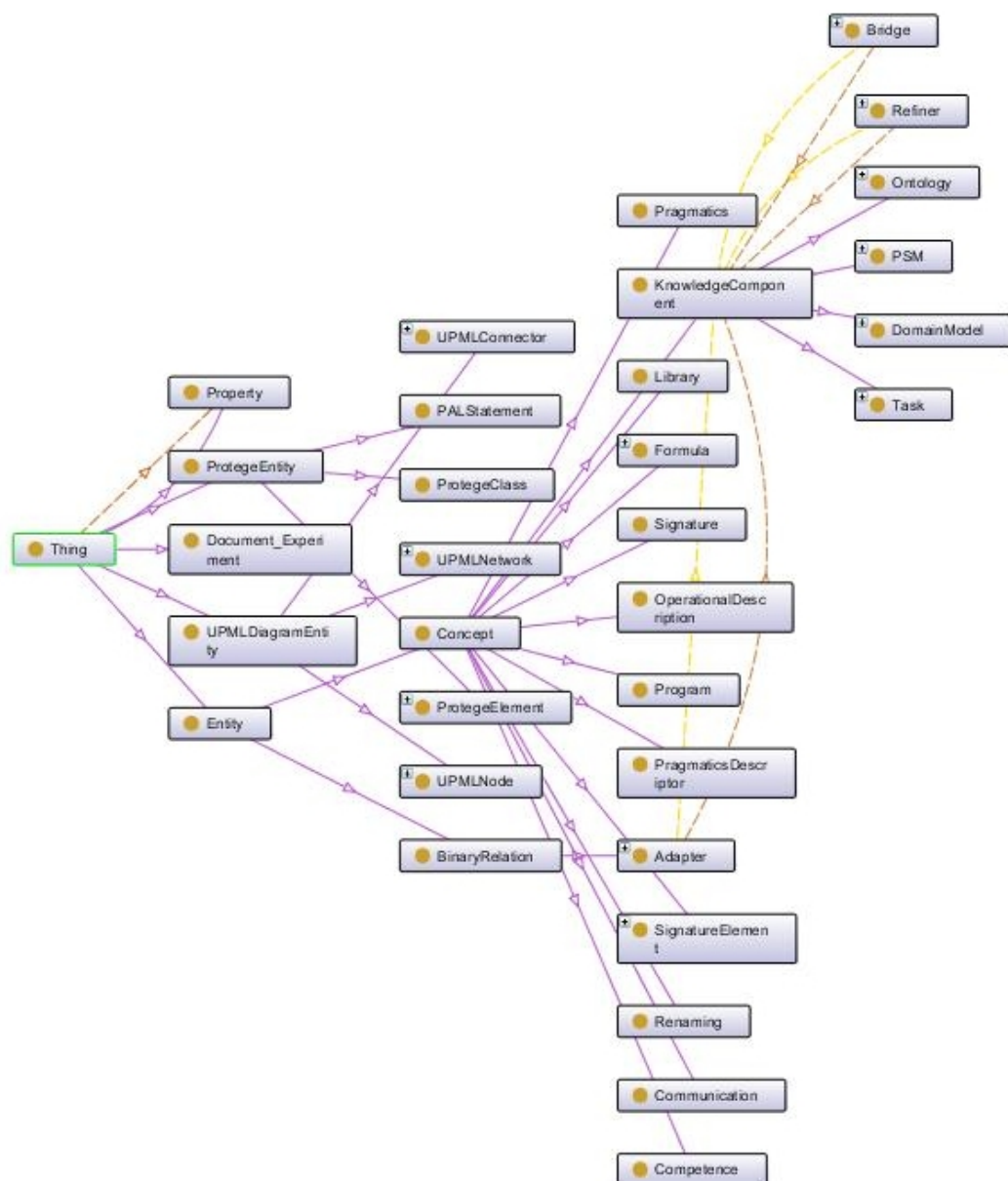
Abbreviation\Acronyms	Description
AAA	Anything about Any
ABox	Assertion Component
ACID	Atomic, Consistent, Isolated and Durable
AUP	Acceptable Use Policy
BDN	Benefits Dependency Network
CA	Customer Agreement
CSA	Cloud Service Agreements
CSCC	Cloud Standards Customer Council
DAML	DARPA Agent Mark-up Language
DL	Description Logics
FoPL	First Order Predicate Logic
FSM	Finite State Machines
HTML	Hypertext Mark-up Language
IaaS	Infrastructure as a Service
IoT	Internet of Things
LSA	Latent Semantic Analysis
NIST	National Institute of Standards and Technology
OIL	Ontology Inference Layer
OWL	Web Ontology Language.
OWL	Web Ontology Language
PaaS	Platform as a Service
PINaaS	Pricing Intelligence as a Service
PSM	Problems Solving Methods
PSM	Problem Solving Method
PSO	Problem Solving Ontology
QoS	Quality of Service.
RBox	Roles/Relationship Component

RDF	Resource Description Framework.
RDF	Resource Description Framework
RE	Requirements Engineering
RuleML	Rule Mark-up Language
SaaS	Software as a Service
SLA	Service Level Agreement
SNOMED	Systematized Nomenclature of Medicine
SWRL	A Semantic Web Rule Language
TBox	Terminological Component
ToS	Terms of Service
UML	Unified Modelling Language
UPML	Universal Problem-solving Method development Language
VDM	Vienna Development Method
VSM	Vector Space Models
WSDL	Web Service Description Language.
WSLA)	Web Service Level Agreement
WSML	Web-service Modelling Language
WSMO	Web Service Modeling Ontology
WSMX	Web Service Execution Language
WWW	World Wide Web

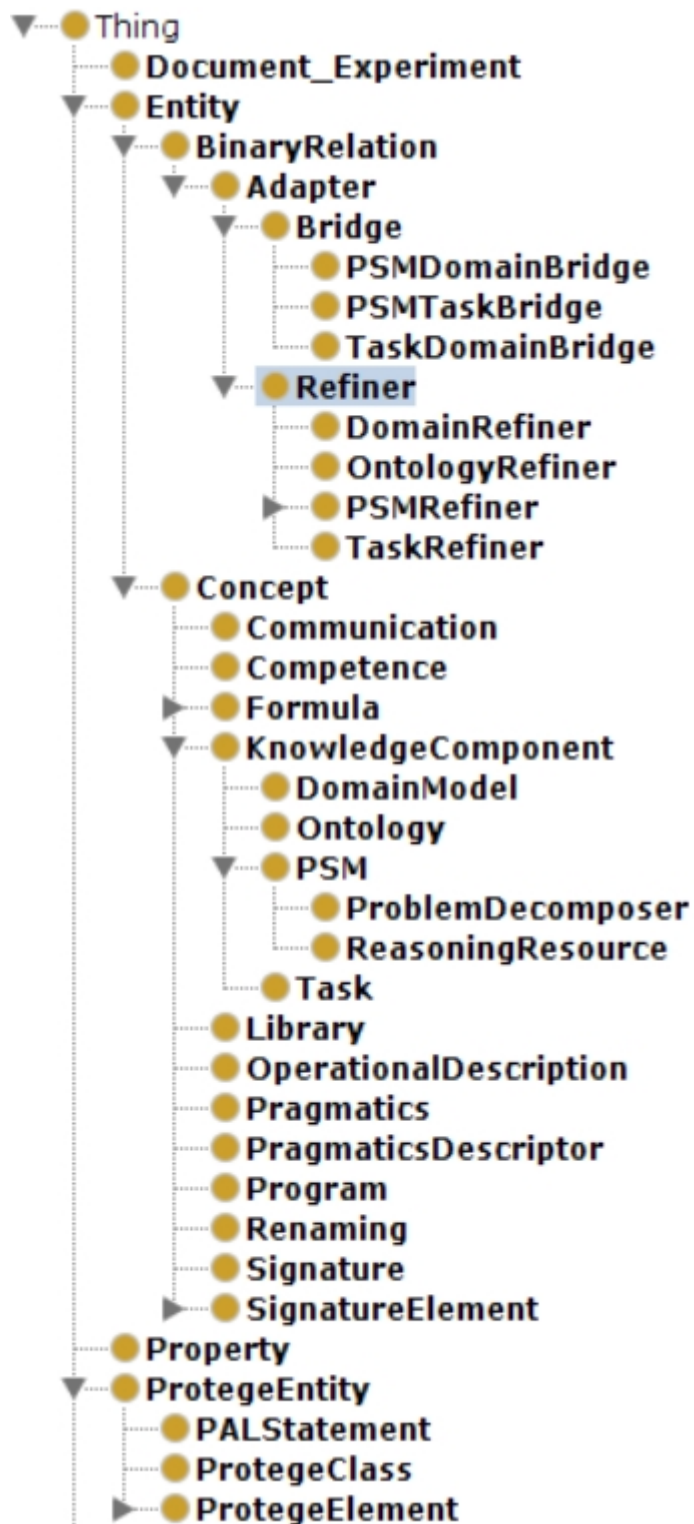


## Appendix B – UPML Ontology

UPML ontology which is used as a base for problem solving ontology development.



Displayed in the OntoGraf Tool



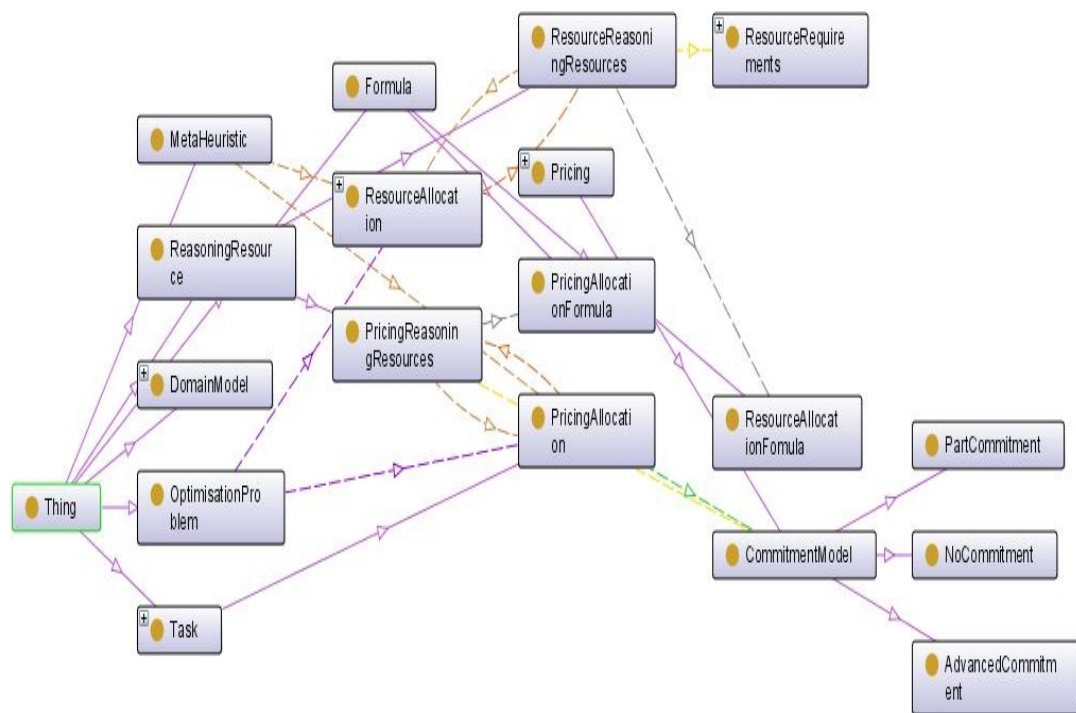
Displayed in Protégé ontology Editor

## Appendix C – An Upper Ontology for Benefits Management of Cloud Computing Investments

The screenshot displays the WebProtégé ontology editor interface. On the left, a hierarchical tree view shows the 'owl:Thing' root class, with 'Action' selected. The tree includes various subclasses such as 'Better', 'New', 'Stop', 'Application', 'Benefit', 'Change', 'Commitment', 'Commitment\_action', 'Competence', 'Degree\_of\_explicitness', 'Disbenefit', 'Driver', 'Enabler', 'Group', 'Objective', 'Organisation', 'Person', and 'Risk'. On the right, the 'Action' class is detailed. The 'URI' field shows the URL 'http://www.semanticweb.org/richard/ontologies/2014/3/benefitsmanagementv1#Action'. The 'Annotations' section lists 'rdfs:label' as 'Action' and 'rdfs:comment' as 'Benefits arise by doing new things, doing things better and stop doing things pg 132.'. The 'Classes' and 'Relationships' sections are currently empty.

An upper ontology for benefits management of cloud computing investments. Displayed in the WebProtégé collaborative ontology editor.

## Appendix D – Hybrid Meta-heuristic Pricing Ontology



Hybrid meta-heuristic pricing ontology. Displayed in the OntoGraf Tool.