UNIVERSITY OF SOUTHAMPTON

Towards a Computable Scientific Method: Using Knowledge Representation Techniques and Technologies to Support Research

by

Mark I. Borkum

A thesis submitted in partial fulfillment for the degree of Doctor of Philosophy

in the
Faculty of Natural and Environmental Sciences
School of Chemistry

August 2013

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF NATURAL AND ENVIRONMENTAL SCIENCES
SCHOOL OF CHEMISTRY

Doctor of Philosophy

by Mark I. Borkum

To conduct research is to actualise the scientific method – the cyclic process of planning and enacting scientific experiments, acquiring data, synthesising information, and organising knowledge. For many years, the primary record of research has been the paper-based laboratory notebook. However, as more researchers are deciding to augment their workflows with the use of software systems, the dominant position of the paper-based laboratory notebook appears to be under threat. In the context of this transition, this thesis explores how knowledge representation techniques and technologies may be used in order to augment the scientific method, and hence, may be used in order to support research.

A core component of this thesis is a detailed consideration of the nature and characteristics of laboratory notebooks, and an exposition of the value proposition for their use by laboratory-based researchers. Three aspects of laboratory notebooks are considered: their capabilities; the nature and characteristics of their content; and, their mediums of implementation.

This thesis argues that, as the majority of the capabilities of a laboratory notebook are generic, the specialisation of a laboratory notebook is a response to the domain-specificity of its content, e.g., its life-cycle, structure, semantics, context, etc. Hence, as the life-cycle of the content is encapsulated by the implementation of the laboratory notebook, and the structure and semantics of the content are derived from the ontology and nomenclature of the domain of discourse, the context of the content of a laboratory notebook must be derived from the circumstances that surrounded its generation, i.e., its provenance. Moreover, as the fundamental difference between the two dominant mediums of implementation (paper and software) is the dis- or collocation of the physical and logical information that constitutes the content, within the context of a laboratory notebook, the qualities of the absence of error (correctness) and conformity to structure and semantics (consistency) are disjoint. Therefore, if sufficient provenance information is captured and curated, then the correctness of consistent content can be determined retrospectively.

Accordingly, this thesis presents techniques and technologies for the machine-processable representation of the prospective and retrospective provenance information of the specification and actualisation of formal processes, such as scientific experiments, recipes and artistic performances – the Planning and Enactment (P&E) ontology. Demonstrating the application of the capture and curation of "intent-centred" provenance information, this thesis describes enhancements to the eCrystals repository for crystal structures, where a partial description of the retrospective provenance of each record is automatically inferred from a prospective description of the formal process for generating a new record.

The P&E ontology is designed to describe the specification and actualisation of formal processes. However, as both specification and actualisation are themselves formal processes, an interesting consequence is that the P&E ontology can be applied recursively. Hence, this thesis presents an exemplar "meta-plan" (a plan for the enactment of another plan; or description of a formal process whose actualisation constitutes the actualisation of another formal process), and outlines how, if such a formal process were encapsulated by a software system, then it would be possible to implement a generic, provenance-aware space, where the retrospective provenance information of events that occur within are (semi-)automatically recorded. Such spaces could then be specialised for specific domains of discourse in order to implement provenance-aware laboratories, kitchens, performance spaces, construction sites, operating theatres, retail environments, etc.

With particular attention to the domains of physical chemistry and crystallography, this thesis also explores how Semantic Web technologies can be used in order to facilitate the implementation of software-based (or electronic) laboratory notebooks and associated software systems. Three new datasets are presented: a controlled vocabulary of quantities, units and symbols that are used in physical chemistry, derived from the subject index IUPAC Green Book; a controlled vocabulary for the classification and labelling of potentially hazardous chemical substances, derived from the Globally Harmonized System of Classification and Labelling of Chemicals (GHS); and, a Linked Data interface for the RSC ChemSpider online chemical database. Demonstrating the use of the GHS dataset, this thesis also presents a Web-based software application, which automates the task of generating Control of Substances Hazardous to Health (COSHH) assessment forms.

Errata Sheet for “Towards a Computable Scientific Method: Using Knowledge Representation Techniques and Technologies to Support Research”

15 September 2014

- After the final submission of this thesis, it was noticed that the subject index of the IUPAC Green Book [99, p. 210] contains a spelling mistake: the term “Hall oefficient” should read “Hall coefficient”. Analysis and discussion in Section 3.1 is not affected by this observation.

Contents

Errata Sheet v

Declaration of Authorship xix

Acknowledgements xxi

1 Introduction 1
- 1.1 Outline 3
- 1.2 Research Contributions 4

2 Background 7
- 2.1 Knowledge Representation Technologies: A Semantic Web Perspective 8
  - 2.1.1 The Semantic Web 8
    - 2.1.1.1 Resource Description Framework (RDF) 10
    - 2.1.1.2 RDF Schema (RDFS) 11
    - 2.1.1.3 Web Ontology Language (OWL) 12
    - 2.1.1.4 Semantic Web Rule Language (SWRL) 13
    - 2.1.1.5 SPARQL Query Language for RDF (SPARQL) 13
  - 2.1.2 Linked Data 14
  - 2.1.3 Commonly-used Vocabularies 15
    - 2.1.3.1 Dublin Core 15
    - 2.1.3.2 Friend of a Friend (FOAF) 16
    - 2.1.3.3 Semantically-Interlinked Online Communities (SIOC) 16
    - 2.1.3.4 OAI Object Reuse and Exchange (OAI-ORE) 17
    - 2.1.3.5 Simple Knowledge Organisation System (SKOS) 19
    - 2.1.3.6 Vocabulary of Interlinked Datasets (VoID) 20
- 2.2 Laboratory Notebooks 21
  - 2.2.1 Nomenclature 24
    - 2.2.1.1 Domain-specific Nomenclature 25
  - 2.2.2 Paper-based Laboratory Notebooks 26
  - 2.2.3 Electronic Laboratory Notebooks 27
    - 2.2.3.1 Characterisation of ELN Content 27
    - 2.2.3.2 Critique 29
      - Collaboratory for Multi-scale Chemical Science (CMCS) 29
      - Prism 30
      - Comb*e*Chem 30
      - eCAT 32
  - 2.2.4 Summary 32

2.3 Applications of Knowledge Representation Techniques and Technologies for Research . . . 33
2.3.1 Computational Workflows . . . 34
2.3.2 Provenance . . . 38
2.3.2.1 Perspectives on Provenance . . . 38
2.3.2.2 Representation of Provenance . . . 40
2.3.2.3 Applications of Provenance . . . 41
2.3.2.4 Critique . . . 42
Open Provenance Model (OPM) . . . 42
PROV . . . 44
Ontology of Scientific Experiments (EXPO) . . . 45
Ontology of Data Mining Investigations (OntoDM) . . . 46
2.3.3 Summary . . . 46

3 Chemistry on the Semantic Web 47
3.1 IUPAC Nomenclature . . . 48
3.1.1 Methodology for Modelling of Subject Indices . . . 50
3.1.2 Methodology for Extraction and Enrichment of Subject Indices . . . 52
3.1.3 Methodology for RDF Representation of Subject Indices . . . 54
3.1.4 Summary of Dataset . . . 55
3.2 Globally Harmonized System of Classification and Labelling of Chemicals (GHS) . . . 60
3.2.1 To Distinguish Between Hazardousness and Riskiness (The Purpose of a Classification And Labelling System) . . . 61
3.2.2 Methodology for Modelling of CLP Regulation . . . 63
3.2.3 Methodology for RDF Representation of CLP Regulation . . . 66
3.2.4 Summary of Web Interface . . . 67
3.2.5 Summary of Dataset . . . 70
3.3 RSC ChemSpider . . . 71
3.3.1 Methodology for Modelling of RSC ChemSpider Records . . . 73
3.3.2 Methodology for Representation of RSC ChemSpider Records . . . 75
3.3.3 Summary of Dataset . . . 78
3.4 Control of Substances Hazardous to Health (COSHH) . . . 80
3.4.1 Legal Implications of Deployment and Use of Automated Artefact Generation Service . . . 81
3.4.2 Value Proposition for Deployment and Use of Automated Artefact Generation Service . . . 84
3.4.2.1 Value Proposition for Individual . . . 85
3.4.2.2 Value Proposition for Organisation . . . 86
3.4.2.3 Value Proposition for Service Provider . . . 86
3.4.2.4 Summary of Value Proposition . . . 87
3.5 Summary . . . 89

4 A Provenance Model for Scientific Experiments 91
4.1 Reflections on Formal Processes . . . 92
4.1.1 Definition of a Formal Process . . . 92
4.1.2 The Act of Description . . . 92
4.1.3 Description of a Formal Process . . . 93

4.1.4 To Distinguish Between Observers and Artefacts 96
4.1.5 Summary 97
4.2 Ontology 98
4.2.1 Entities and Relationships 98
4.2.1.1 Generation 102
4.2.1.2 Utilisation, Modification and Destruction 102
4.2.1.3 Causation and Lineage 102
4.2.1.4 Actualisation and Satisfaction 103
4.2.2 Life Cycles 104
4.2.2.1 Artefacts 105
4.2.2.2 Actions 106
4.2.2.3 Enactments 107
4.2.3 Assumptions 107
4.2.3.1 The Enactment Environment (Space) 108
4.2.3.2 Agents are Artefacts 109
4.2.3.3 Annotations are Artefacts 109
4.2.3.4 Reification of Retrospective Relationships 110
4.2.4 Summary 111
4.3 Integration with eCrystals Repository for Crystal Structures 113
4.4 Plan for the Enactment of Plans 115
4.5 Summary 118

5 Conclusions **125**
5.1 Research Outcomes 129
5.2 Further Work 130

A Contents of CD-ROM **133**

Bibliography **135**

List of Figures

2.1 The Semantic Web Stack (available at http://www.w3.org/2007/03/layerCake.png - Copyright W3C). 9

2.2 Exemplar SPARQL query that uses terms from the FOAF vocabulary in order to select the name and mailbox ("mbox") of each agent that is described by the RDF data source. 14

2.3 Depiction of the entity-relationship model for the SIOC Core Ontology (figure taken from http://www.w3.org/Submission/2007/SUBM-sioc-spec-20070612/). 17

2.4 The aggregation A-1 aggregates three resources and is described by resource map ReM-1 (figure and caption taken from http://www.openarchives.org/ore/1.0/primer.html). 18

2.5 Depiction of relationships between conceptual entities that collectively encapsulate five of the seven W's of provenance. 39

2.6 Depiction of entities and relationships in the core ontology for the Open Provenance Model Vocabulary (OPMV) (available at: http://open-biomed.sourceforge.net/opmv/img/opmv_main_classes_properties_3.png) . 44

2.7 Depiction of the organisation of PROV specifications (available at: http://www.w3.org/TR/2013/WD-prov-overview-20130312/prov-family.png) 44

3.1 Excerpt of the subject index of the third edition of the IUPAC Green Book (left) and corresponding LaTeX source (right). 50

3.2 UML class diagram for the proposed entity-relationship model of the subject index of the third edition of the IUPAC Green Book. 51

3.3 Non-terminal production rules of a grammar (in ANTLR v3 syntax) whose corresponding parser recognises indices that were generated by the theindex environment for LaTeX. 52

3.4 Depiction of RDF graph that describes three terms from the subject index of the IUPAC Green Book. 55

3.5 Distribution of references in the subject index of the third edition of IUPAC Green Book. 56

3.6 Histogram of total number of references to pages in the subject index of the third edition of the IUPAC Green Book. 57

3.7 Histogram of total number of references to terms in the subject index of the third edition of the IUPAC Green Book. 58

3.8 Depiction of weighted list (or "tag cloud") of most frequently referenced terms in the subject index of the third edition of the IUPAC Green Book. 59

3.9 Depiction of the riskiness of a hazard whose risk assessment function is defined by a linear gradient. The "traffic light" colouring system is used to indicate the magnitude of the codomain at each point. 62
3.10 Depiction of RDF schema for core GHS entities and their inter-relationships (some entities and relationships are not shown). 63
3.11 Depiction of RDF graph that describes the hazard category "Flammable solid; category 1", along with its associated hazard class and pictogram. 66
3.12 Depiction of RDF graph that describes the chemical substance "hydrogen", along with its associated classification and labelling entities. 67
3.13 Exemplar SPARQL query that locates all substances, whose IUPAC name contains the string "aluminium". 68
3.14 Screen shot of GHS web application; list of hazard pictograms (available at: http://ghs.chem.soton.ac.uk/hazard_pictograms). 69
3.15 Screen shot of GHS web application; description of substance: "aluminium lithium hydride" (available at: ghs.chem.soton.ac.uk/substances/1). 70
3.16 Depiction of the directed graph of "InChI" Web services provided by RSC ChemSpider, where nodes denote chemical identifier formats, and edges denote the availability of a Web service that provides an injective and non-surjective mapping for chemical identifiers from the source to the target format. 72
3.17 Depiction of RDF graph that describes the compound "Water" using terms from the ChemAxiom ontology (available at: http://www.chemspider.com/Chemical-Structure.937.rdf). 76
3.18 Depiction of OAI-ORE aggregation of information resources associated with an RSC ChemSpider record. 76
3.19 Activity diagram that describes the dereferencing of a URI, to give an OAI-ORE aggregation for an RSC ChemSpider record. 77
3.20 Activity diagram that describes the successful resolution of a chemical identifier, and dereferencing of the obtained URI, to give an OAI-ORE aggregation for an RSC ChemSpider record. 77
3.21 Activity diagram that describes the unsuccessful resolution of a chemical identifier. 78
3.22 Depiction of RDF graph that asserts the relationships between the RSC ChemSpider and OpenMolecules descriptions of the compound "Methane" (available at: http://rdf.openmolecules.net/?InChI=1/CH4/h1H4). . 78
3.23 Screen shot of COSHH assessment form generated from GHS description of substance: "aluminium lithium hydride" (index number: 001-002-00-4). 82
3.24 Depiction of the relationships between the three considered parties (the individual, organisation, and service provider), and the service that is being provided. 85

4.1 Depiction of a prospective description of a formal process (ellipses and rectangles denote artefacts and actions respectively), where a bomb will be ignited, and will subsequently explode. 96
4.2 Depiction of a prospective description of a formal process (ellipses and rectangles denote artefacts and actions respectively), where a sensor will use sense perception in order to generate data. 97

4.3 UML class diagram for the Planning and Enactment (P&E) ontology. . . 99
4.4 Depiction of asserted and inferred relationships between entities in an excerpt of a prospective description of a formal process (ellipses and rectangles denote artefacts and actions respectively). 101
4.5 Depiction of the life-cycle of an artefact, as described by the Planning and Enactment (P&E) ontology (where ϵ denotes an epsilon transition). 105
4.6 Depiction of the life-cycle of an action, as described by the Planning and Enactment (P&E) ontology (where ϵ denotes an epsilon transition). . . . 106
4.7 UML class diagram for an extension to the Planning and Enactment (P&E) ontology, which defines the concepts of the enactment environment (a space) and location. 108
4.8 UML class diagram for an extension to the Planning and Enactment (P&E) ontology, which defines the concept of an agent. 109
4.9 UML class diagram for an extension to the Planning and Enactment (P&E) ontology, which defines the concept of an annotation. 109
4.10 Depiction of asserted and inferred relationships between entities in an excerpt of a retrospective description of a formal process (ellipses, rectangles and octagons represent artefacts, actions and reifications respectively). . 111
4.11 Depiction of the Planning and Enactment (P&E) ontology. Nodes representing classes and predicates are coloured grey and white respectively. 112
4.12 Depiction of the prospective description of the eCrystals crystal structure determination workflow, described in terms of the oreChem Core Ontology (the precursor to the Planning and Enactment (P&E) ontology). Rectangles and ellipses correspond to software applications and data files respectively. Available at: http://ecrystals.chem.soton.ac.uk/plan.rdf 120
4.13 Depiction of the retrospective description of the partial enactment of the eCrystals crystal structure determination workflow, for record #29, where rectangles and ellipses correspond to software applications and data files respectively, and solid and dashed edges correspond to assertions of the orechem:emitted and orechem:used predicates. Available at: http://ecrystals.chem.soton.ac.uk/cgi/export/29/ORE_Chem/ecrystals-eprint-29.xml 121
4.14 SPARQL query that returns a set of quads, where each quad includes a reference to a retrospective description of the enactment of a formal process, along with references to the raw, intermediate, and reported data files that were used and/or generated during said enactment. 122
4.15 Depiction of the retrospective description of the partial enactment of the eCrystals crystal structure determination workflow, for record #29, where each ellipse corresponds to a data file, and edges correspond to assertions of the orechem:derivedFrom predicate. Available at: http://ecrystals.chem.soton.ac.uk/cgi/export/29/ORE_Chem_Provenance/ecrystals-eprint-29.png 122
4.16 Depiction of the flow diagram for a plan that describes the realisation of another plan (where ϵ denotes an epsilon-transition). 123

List of Tables

2.1 Characterisation of the content of ELN entries, given the presence or absence of structure and semantics. 28

3.1 Namespaces and prefixes used in Section 3.1.3. 54
3.2 Terms from the subject index of the third edition of the IUPAC Green Book with 10 or more references (terms with the same frequency are given in alphabetical order). 58
3.3 Namespaces and prefixes used in Section 3.2.3. 66
3.4 Instances of model entities in GHS dataset. 70
3.5 Namespaces and prefixes used in Section 3.3.2. 75
3.6 Cost-benefit analysis for the deployment and utilisation of an automated artefact generation service, e.g., a health and safety assessment form generator. 88

4.1 Namespaces and prefixes used in Section 4.2. 98
4.2 Relationships between artefacts and actions in the Planning and Enactment (P&E) ontology. Additional properties of each relationship are given in parenthesis, where: (†) denotes being functional; (‡) denotes being inverse functional, and (§) denotes transitivity. 100

List of Algorithms

1 Binary function for normalisation of labels for terms that are extracted from the subject index of the third edition of the IUPAC Green Book. . 53

Declaration of Authorship

I, **Mark Ian Borkum**

declare that this thesis

> **Towards a Computable Scientific Method: Using Knowledge Representation Techniques and Technologies to Support Research**

and the work presented therein are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University;
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- Where I have consulted the published work of others, this is always clearly attributed;
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- Parts of this work have been published as:

M. Borkum, C. Lagoze, J.G. Frey and S.J. Coles, A semantic eScience platform for chemistry, in *Proceedings of the 6th IEEE International Conference on e-Science*, IEEE Computer Society, 2010.

M. Borkum, S.J. Coles, and J.G. Frey, Integration of oreChem with the e-Crystals repository for crystal structures, in *Proceedings of the 9th UK e-Science All Hands Meeting*, 2010.

C. Lagoze, P. Mitra, W.J. Brouwer and M. Borkum, The oreChem project: Integrating chemistry scholarship with the Semantic Web and Web 2.0, in *Proceedings of the Microsoft eScience Workshop*, Microsoft Research, 2009.

Signed: ....

Date: ....

Acknowledgements

To my supervisors, Jeremy Frey and Simon Coles, for whom no amount of thanks is too much. While we may sometimes disagree, our discussions are always profitable. Your continual support and advice have helped me to stay on track, keep my focus, and work at a steady pace. I would also like to express my sincerest appreciation to Colin Bird, whose rational approach to eliciting rationale benefited me enormously. To David De Roure and Carole Goble, thank you for trusting me, and for giving me the profound opportunity to contribute to the myExperiment project. No-one could have predicted that the Physical Chemist whom I would meet by chance, while giving a demonstration of my contributions at a conference in Nottingham, would later supervise my PhD. To the anonymous Austrian gentleman, who sat next to me on the train from Southampton Central to London Waterloo, and suggested that I purchase a book on a then unknown software development framework called “Woo-bee on Whales” (Ruby on Rails), I offer my deepest gratitude. One month later, and thirty pounds (sterling) lighter, I confidently stood up, and announced to the attendees of a meeting in Manchester that I would single-handedly redesign one of their projects’ deliverables. Audacious behaviour, which would be rewarded with a trip to Nottingham. Last, but by no means least, to my parents, Hilda and Steven, thank you for everything (I love you both very much).

Dedicated to the memory of J. Lee Dirks, friend and colleague.

Chapter 1

Introduction

"In theory, there is no difference between theory and practice.
But, in practice, there is." – Manfred Eigen
(also attributed to Lawrence Peter "Yogi" Berra, Chuck Reid,
Jan L. A. van de Snepscheut, et al.)

To propose a theory, is to specify a set of principles; a plan, which may be used in order to guide the enactment of a sequence of actions. However, as enunciated by the quotation, no matter how meticulously we devise our plans, or how zealously we follow our own instructions, the outcome of the enactment of a plan rarely (if ever) satisfies our original intentions.

Fortunately, all is not lost. Indeed, if events always went according to plan, then mankind would not have enjoyed some of its most significant scientific advances. For example, until Galileo Galilei conducted his famous experiment, it was commonly believed that, when dropped simultaneously from a given height, a stone would always hit the ground before a feather. Similarly, had Leo Baekland been less prone to slamming the door to his laboratory, he may not have discovered the thin film of polyoxybenzylmethyleneglycolanhydride (more commonly known as Bakelite) that had formed on the wooden shelf of his chemicals cabinet.

As we well know, the history of science is not the exclusive result of chance and serendipity. Instead, it is a consequence of the deep thought of many gifted individuals, and the rigorous application of the scientific method; as exemplified by Galileo himself. However, it is useful to consider how the aforementioned events could have transpired, had their participants been using computers, i.e., had their experiments been conducted, and the outcomes of said experiments been recorded, with the assistance of knowledge representation techniques and technologies, software applications, and associated services.

With tongue in cheek, we suggest that, as he was a busy individual, Galileo would most likely have decided to commission a local company to develop a bespoke software application, tailored specifically to his needs. Unfortunately, this decision would prove to be unwise. This is because, given the current sum of all scientific knowledge, it would have been obvious to the software developers that different "droppable things" would take different amounts of time to fall to Earth. Codifying this "fact", it would have been sensible for said software developers to use the calculated rate of acceleration as a unique identifier (sadly, these software developers would have had very poor judgement!) Unfortunately, their approach would have had a serious drawback. Namely that, if two "droppable things" were observed to possess the same acceleration, then they would both be allocated the same, supposedly unique, identifier, and hence, the true context of measurements would be lost. To the dismay of the software developers, shortly after discovering that the rate of acceleration of all bodies towards the centre of the Earth is everywhere constant, the client would have returned the software application, and demanded his money back.

Similarly, if Baekland were using a bespoke software application, then how would he have recorded his totally unexpected observation? The field-sets in his software application would have been fixed for a specific set of experiments. Hence, given an unexpected observation of an event that did not transpire during the enactment of an experiment, he would have been forced to either: select an arbitrary experiment, and record his observation using the "least inappropriate" field; or, forego the recording of the observation entirely. Unfortunately, this approach would also have a serious drawback. Namely that, even though it may be possible to persist a warped and untrue representation of the observation, the context, structure and semantics (meaning) of the observation are lost. In his frustration, Baekland would resort to using a second software application to record his observation. However, now he would need a third software application, in order to record the association between the two original datasets. He would think to himself, "data integration is too complicated!"

Obviously, the two scenarios are highly contrived. However, the fictional software applications serve to illustrate two salient points. First, that the context, structure and semantics of the specification of a plan are as important as the specifications themselves. It must be possible to deviate from a given plan, and to improvise in the absence of a plan. Second, that when theory is different from practice, the use of knowledge representation techniques and technologies, and to an extent, the use of computers in general, does not necessarily enhance our ability to conduct research.

In conclusion, the purpose of this thesis is to explore how knowledge representation techniques and technologies may be used in order to augment the scientific method, and hence, may be used in order to support research.

1.1 Outline

In the following, we outline the structure and content of the remainder of this thesis.

In Section 2.1, we introduce the concept of a knowledge representation technology, and discuss specific examples. In this section, we pay particular attention to techniques that are relevant to the representation of knowledge that is gained as part of scientific research, e.g., techniques for asserting ownership and scholarship; for communicating domain-specific terminology; and, for aggregating collections of heterogeneous resources, such that they may be encapsulated and disseminated as a single, logical unit.

An integral part of the scientific method is the persistence of rationale, methodology, observations and outcomes. Hence, in Section 2.2, we explore concept of a laboratory notebook – the primary record of research. Specifically, we discuss the characteristics of laboratory notebooks, paying particular attention to the notion of domain-specificity and the act of specialisation, i.e., we consider both the similarities and differences between laboratory notebooks that are specialised for distinct domains of discourse. Moreover, we attempt to understand the process by which a paper-based laboratory notebook is digitised, i.e., realised as a software system. Finally, we critique exemplar electronic laboratory notebook systems, including: the Collaboratory for Multi-scale Chemical Science (CMCS), Comb*e*Chem, and eCAT.

In Section 2.3, we identify potential applications of knowledge representation techniques and technologies, which are applicable to research. In particular, we discuss the concept of a computational workflow – a specification of a formal process that encapsulates the sequence of actions for a software-assisted computation – and of provenance information – data that asserts the lineage (or origins) of other data. Moreover, we critique existing mechanisms for the exposition of both prospective and retrospective provenance information, including: the Open Provenance Model (OPM), PROV and the Ontology of Scientific Experiments (EXPO).

Given this background, in Chapter 3, we describe in detail a suite of domain-specific datasets, and corresponding software applications and services. Aside from the discussion each dataset, we attempt to use the process of the design and implementation of each deliverable in order to gain an understanding of the wants and needs of the laboratory-based researcher, i.e., to learn the value-proposition for the use of software systems in laboratory environments. Specifically, in Section 3.1, we develop a machine-processable representation of a domain-specific nomenclature; in Section 3.2, we develop a machine-processable representation of a generic nomenclature for the exposition of classification and labelling information for hazardous chemical substances; in Section 3.3, we enhance a popular, online chemical database, such that each record is accompanied by a machine-processable description; and, in Section 3.4, we utilise our new datasets in order to implement an automated service for the generation of health and safety assessment forms.

Thus far, we have hinted at the idea of regarding a scientific experiment as being comparable to a description of a formal process. In Section 4.1, we reflect on this comparison, and offer a more detailed consideration of the nature and characteristics of formal processes, and their applicability to the representation of scientific research. We give focus to what is meant by the concept of a "formal process" and the act of description (of a formal process), paying particular attention to the assumptions that influence subsequent logical deductions and inferences, including: the characteristics that enable the distinguishing of active and passive participants, and of human beings and automata. In Section 4.2, we codify our reflections as a machine-processable ontology for the planning and enactment of formal processes. This is followed by Section 4.3, where we evaluate our approach, by enhancing the eCrystals repository for crystal structures, with a machine-processable description of the retrospective provenance information for each record in the database. Finally, in Section 4.4, we outline the design and implementation of a meta-plan, i.e., a plan that describes the enactment of other plans, which is intended to support the actualisation of a provenance-aware space.

The thesis concludes in Chapter 5, where we summarise our main results, and identify outcomes that are worthy of further investigation.

1.2 Research Contributions

This thesis has made the following novel research contributions:

1. The design of a controlled vocabulary for quantities, units and symbols that are used in physical chemistry.
2. The design of a controlled vocabulary for the classification and labelling of potentially hazardous chemical substances.
3. The design of a Linked Data interface for the RSC ChemSpider online chemical database.
4. The implementation of a fully automated, legally compliant, Web-enabled service for the generation of health and safety assessment forms.
5. Consideration of the nature and characteristics of formal processes. In particular, the philosophical and technical implications of modelling assumptions.
6. The design of an ontology for the exposition of both the prospective and retrospective provenance of formal processes.

7. The specification of a formal process for the enactment of another formal process (that is described in terms of the ontology).

8. The implementation of a Linked Data interface for the eCrystals repository for crystal structures.

The third contribution was designed in consultation with the RSC ChemSpider team, which includes: Colin Batchelor, Richard Kidd, Jonathan Steele, and Antony Williams.

The foundation for the sixth contribution was laid as part of the oreChem project, whose contributors included: Simon Coles, J. Lee Dirks, Jim Downing, C. Lee Giles, Geoffrey Fox, Jeremy Frey, Carl Lagoze, Prasenjit Mitra, Karl Mueller, Peter Murray-Rust, Marlon Pierce, Theresa Velden, and Alex Wade. Following the termination of the oreChem project, work was continued independently.

The eighth contribution was designed in collaboration with members of the EPSRC UK National Crystallography Service (NCS), which includes: Simon Coles, Peter Horton, and Graham Tizzard.

Chapter 2

Background

To conduct research is to enact the scientific method, to generate data, and to synthesise information and knowledge. Facilitated by recent advancements in pervasive computing and telecommunication technologies, the nature of the scientific method is undergoing a transition. Originally an isolated venture of individuals, who periodically communicated with each other, disseminating their research outcomes as monolithic "scholarly works", the scientific method is becoming a globally-distributed, decentralised protocol for the production and consumption of data, where researchers are in continuous communication, disseminating any and all aspects of their own research, including the content of their laboratory notebooks, using the Web, with varying degrees of encapsulation and granularity.

However, the transition towards continuous communication is not a silver bullet. Ever-present and difficult problems, such as the determination of logical correctness, and the detection of fraud, are not resolved by simply digitising the data, and making it accessible via the Web. In order for the data to be produced and disseminated, and subsequently, consumed and manipulated, in a consistent and unambiguous manner, it is necessary for said data to be represented in a structured and machine-processable form, and for said structure to be interpretable according to a semantics.

In this chapter, we introduce technologies for the machine-processable representation of data. These technologies, and the methodology for their selection, are of vital importance, and worthy of consideration, as they directly affect the future capabilities for data dissemination and reuse. Given this context, we proceed to characterise the data that is being produced and consumed during research, i.e., the contents of the laboratory notebook, with the goal of understanding how the aforementioned technologies may be applied, in order to enhance said research.

2.1 Knowledge Representation Technologies: A Semantic Web Perspective

The purpose of a knowledge representation technology is to provide a mechanism for encoding descriptions of conceptual entities as symbolic messages, which may be communicated with other users of the same technology, and manipulated, by following the rules of a calculus.

Typically, our (humanity's) motivation for using knowledge representation technologies is to communicate with machines. Indeed, encoding descriptions of conceptual entities as symbolic messages has three notable advantages. First, and foremost, the content of each message is wholly determined by that of its symbols. Hence, communication via knowledge representation technologies is unambiguous, with respect to the content of each message[1]. Second, the content of a message may be manipulated by naïve automata, which are agnostic to the interpretation of each symbol. Thus, knowledge representation technologies enable the emulation of human processes, such as logical inference. Finally, messages may be persisted, with fixity, for future reuse. Therefore, the use of knowledge representation technologies facilitates cognitive delegation, and provisions for the realisation of ephemeral value.

However, it should be noted that a machine neither "understands" the meaning of a message, nor does it "infer" the content of new messages. Instead, the capabilities of a machine, to recognise and/or manipulate the symbols of a message, are completely defined by the structure and semantics of the knowledge representation technologies that are being used. Thus, we may refer to messages as "machine-readable" or "machine-processable", but not as "machine-understandable" or "machine-inferable". Furthermore, the level of sophistication of the aforementioned capabilities is measured on a continuous, rather than discrete, scale, e.g., it is possible to recognise the basic relationships that are asserted by descriptions of conceptual entities, the patterns that predict said relationships, the underlying principles that govern said patterns, etc. Therefore, we may characterise knowledge representation technologies by the sophistication of their recognition and manipulation capabilities.

2.1.1 The Semantic Web

The Semantic Web is a political movement, which argues for the inclusion of machine-processable data in hypertext Web documents [1]. The goal of the Semantic Web movement, is to convert the information content of unstructured and semi-structured hypertext Web documents into a "Web of data" [2]. The activities of the Semantic

[1] No claims are made about the ambiguity of the interpretation of the content of each message.

Web movement, which include the specification and implementation of new technologies, and the exposition of best practice, are coordinated by the World Wide Web Consortium (W3C)[2].

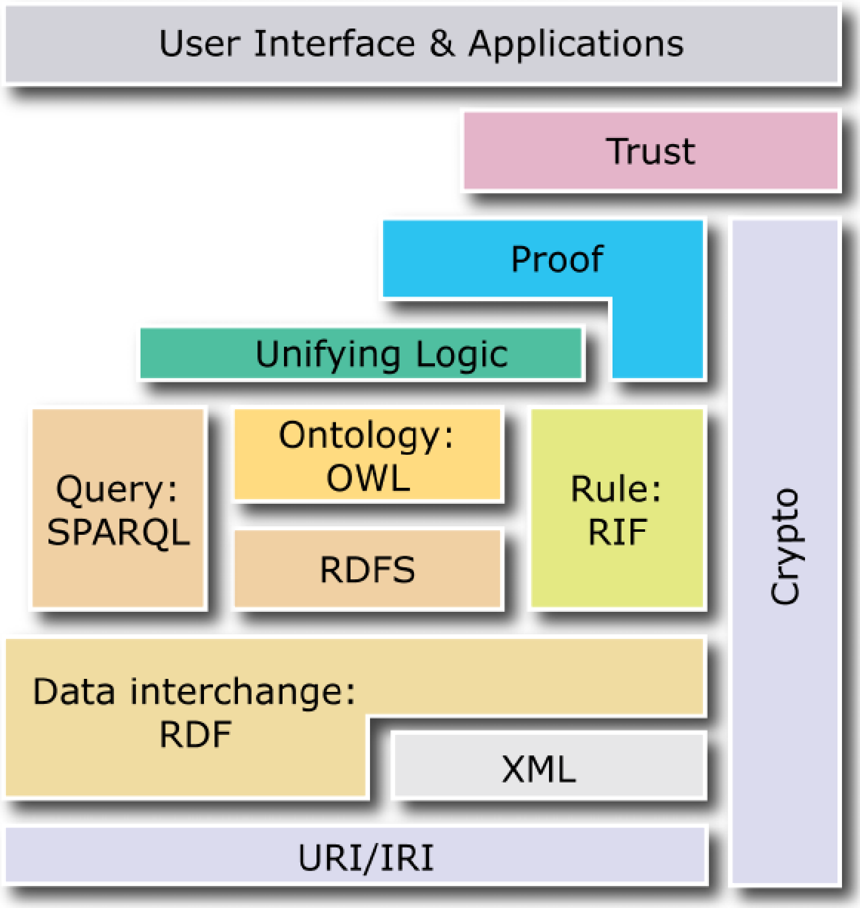


FIGURE 2.1: The Semantic Web Stack (available at http://www.w3.org/2007/03/layerCake.png - Copyright W3C).

The architecture of the Semantic Web, commonly referred to as the "stack" or "layer cake", is depicted in Figure 2.1. Each successive layer builds on the functionality that is provided by the previous layers, and consists of one or more technologies, where the specifications for each technology have been ratified by the W3C. At the time of writing, the technologies to support the upper-most layers have yet to be standardised.

In the remainder of this section, we describe each layer of the Semantic Web Stack, paying particular attention to the technologies that are relevant to the work that is presented in this thesis. In Section 2.1.1.1, we describe the recommended format for data interchange on the Semantic Web, the Resource Description Framework (RDF). In Section 2.1.1.2, we describe an extension of RDF, which provides an enhanced vocabulary for data description, RDF Schema (RDFS). In Section 2.1.1.3, we describe the Web Ontology Language (OWL), which provides the basis for ontology on the Semantic Web. This

[2] http://www.w3.org/

is followed by Section 2.1.1.4, which introduces Semantic Web Rule Language (SWRL), a technology for describing forward-chaining inference rules. Finally, in Section 2.1.1.5, we introduce the query language for the Semantic Web, SPARQL Protocol and RDF Query Language (SPARQL).

2.1.1.1 Resource Description Framework (RDF)

Resource Description Framework (RDF) is a family of specifications [3, 4], which collectively define a methodology for the modelling and representation of information resources as structured data.

In RDF, the fundamental unit of information is the subject-predicate-object 3-tuple, which is referred to as the "RDF triple" or "triple". Each triple encapsulates the assertion of a single proposition, where: the "subject" denotes the source of the assertion; the "object" denotes the target of the assertion; and, the "predicate" is a verb, which relates the source to the target.

The elements of an RDF triple are either resources or literal values, referred to as "RDF labels" or "labels". Resources are identified by either a Uniform Resource Identifier (URI) reference, which indicates the existence of a named thing, i.e., a thing whose name is asserted; or, a blank node, which indicates the existence of a thing, about whom nothing is asserted. Literals are either typed or untyped (plain). The "subject" and "predicate" of a triple must both be resources. The "object" of a triple may be either a resource or a literal value.

For example, the natural-language sentence

$$\text{The } \underbrace{\text{capital city of}}_{predicate} \text{ the } \underbrace{\text{United Kingdom}}_{subject} \text{ is } \underbrace{\text{London}}_{object}.$$

can be encoded[3] as the following RDF triple

$$\underbrace{\mathtt{dbpedia} : \mathtt{United\_Kingdom}}_{subject} \; \underbrace{\mathtt{dbpedia} - \mathtt{owl} : \mathtt{capital}}_{predicate} \; \underbrace{\mathtt{dbpedia} : \mathtt{London}}_{object}$$

where `dbpedia:United_Kingdom` and `dbpedia:London` are qualified names that denote resources from the DBPedia dataset [5], which describe the conceptual entities "United Kingdom" and "London" respectively; and, `dbpedia-owl:capital` is a qualified name that denotes a predicate from the accompanying DBPedia ontology[4].

[3] Since many vocabularies, taxonomies and ontologies may exist for a given domain of discourse, it is possible for an assertion to have many encodings.

[4] http://dbpedia.org/ontology/

In RDF, the fundamental unit of communication (for the exchange of information) is the unordered set of triples, which is referred to as the “RDF graph” or “graph”. We note that the set of all RDF graphs is an abelian group, where the identity element is the empty graph (the singleton set of zero triples), and the associative binary operation is set union. Hence, any two graphs may be combined to yield a third graph. However, while the semantics of combination are essentially monotonic[5], it is important to note that characteristics such as logical consistency do not necessarily distribute over the group operation, i.e., while two graphs may be independently consistent, their combination may be inconsistent.

Many syntaxes are available for the serialisation of RDF graphs. The most popular syntaxes are either text-based, e.g., Notation 3 (N3) [6], N-Triples [7], and Turtle [8]; or, are defined as transformations from the RDF abstract data model to that of another system, e.g., RDF/XML [9] and JSON-LD [10]. We note that, as the set of all RDF graphs is a monoid, the set of all serialisations of RDF graphs for a given format is also a monoid.

2.1.1.2 RDF Schema (RDFS)

RDF Schema (RDFS) is a self-hosted extension of RDF, which defines an RDF vocabulary for the description of other RDF vocabularies [11].

RDFS extends the RDF data model, by providing metadata terms for the description and instantiation of basic entity-relationship models. Hence, RDFS may be used in order to give additional structure to RDF data, e.g., by restricting the domain and/or codomain of a relationship to instances of a specific class of entities, using the `rdfs:domain` and `rdfs:range` predicates. Moreover, RDFS may be used in order to give additional structure to the entities and relationships themselves, e.g., by asserting arbitrary hierarchies using the `rdfs:subClassOf` and `rdfs:subPropertyOf` predicates.

Like RDF, the RDFS specification defines an entailment regime for well-formed graphs [4], where new triples may be automatically inferred from existing ones, by applying each member of a set of production rules, e.g., both the transitive and reflexive closures of the two hierarchical relationships are automatically inferred, such that: t_n is a sub-class of t_n (reflexivity); and, if t_1 is a sub-class of t_2, and t_2 is a sub-class of t_3, then t_1 is also a sub-class of t_3 (transitivity).

However, it should be noted that RDFS has two key limitations, which restrict the scope of its semantics. First, neither RDFS, nor its “sibling” RDF, specify metadata terms for the assertion of characteristics of predicates, which may facilitate enhanced reasoning

[5] When two RDF graphs are combined, in this case using an infix binary operator, the assertions of the “right” graph do not overwrite or supersede those of the “left” graph. Instead, the assertions exist simultaneously.

about said predicate, such as: transitivity, reflexivity, or symmetry. Consequentially, the entailment regime for RDFS is defined explicitly in terms of specific predicates. Second, RDFS does not incorporate any aspects of set theory, such as taking the intersection or union of the set of instances of specific classes. Hence, RDFS is not capable of describing certain types of RDF vocabularies, such as those that contain disjoint classes.

2.1.1.3 Web Ontology Language (OWL)

Web Ontology Language (OWL) extends the RDFS data model by providing additional metadata terms for the description and instantiation of arbitrarily complex entity-relationship models [12]. However, given the inherent variability in the complexity of entity-relationship models, OWL is available as three different "species" (sub-languages), where successive species are used in order to describe increasingly complex models: OWL Lite, OWL DL, and OWL Full.

OWL Lite has the expressiveness of the $\mathcal{SHIF}^{(\mathcal{D})}$ description logic. The purpose of OWL Lite is to provide a "light" version of OWL DL, which is suitable for third-party software developers, who wish to support OWL in their software systems. OWL Lite uses the language constructs of RDFS in order to specify the `owl:Class` class, along with two additional classes, `owl:DatatypeProperty` and `owl:ObjectProperty`, whose instances describe the characteristics of literal and resource-valued predicates respectively, e.g., in OWL Lite, predicates may be transitive, symmetric, functional, or inverse functional; or, the inverse of other predicates. Moreover, OWL Lite includes language constructs to assert the equivalence and/or disjointedness of specific classes and predicates, or of individual instances.

OWL DL has the expressiveness of the $\mathcal{SHOIN}^{(\mathcal{D})}$ description logic. The purpose of OWL DL is to provide a maximally-restricted subset of OWL Full language constructs, whilst ensuring that a decidable reasoning procedure can exist for an OWL reasoner, e.g., in OWL DL, one may assert cardinality constraints for any predicate, providing that neither said predicate, its inverses, nor super-predicates, are transitive.

Finally, OWL Full contains all OWL language constructs, and provides free, unconstrained use of RDF and RDFS constructs. The key difference between OWL Full and OWL DL is that, in OWL Full, the resource `owl:Class` is equivalent to the resource `rdfs:Class`, whereas, in OWL DL, it is a proper subclass. Hence, in OWL DL, not all RDFS classes are OWL classes. The main implication of this difference is that, at the cost of being neither logically sound nor complete, OWL Full provides far more flexibility than OWL DL. Hence, it is recommended [13] that OWL Full should only be used when it is impossible to describe the domain using OWL DL.

2.1.1.4 Semantic Web Rule Language (SWRL)

Semantic Web Rule Language (SWRL) [14] is based on a combination of OWL DL and OWL Lite with the Rule Markup Language (RuleML) [15]. The purpose of SWRL is to extend the set of OWL axioms to include Horn-like rules, with the goal of enabling said rules to be combined with the assertions of pre-existing OWL knowledge bases, in order to infer new assertions, which could not otherwise have been made, given only the language constructs and semantics of OWL.

In SWRL, each rule is a combination of an antecedent (body) and a consequent (head), which is interpreted as a logical implication, i.e., if and only if the conditions that are specified by the antecedent hold, then the conditions that are specified by the consequent hold. Both the antecedent and consequent may consist of zero or more logical atoms, which are related by logical conjunction. Hence, the set of zero atoms is always holds.

The atoms are defined as follows:

C (x) – Denotes the membership of the resource x in OWL class C.

P (x, y) – Denotes the assertion of the OWL predicate P for resources x and y.

sameAs (x, y) – Denotes the assertion of the `owl:sameAs` predicate for resources x and y.

differentFrom (x, y) – Denotes the assertion of the `owl:differentFrom` predicate for resources x and y.

f $(x, \dots)$ – Denotes the application of the built-in function f to the arguments x, etc. It should be noted that the set of built-ins for SWRL is version specific, and subject to change in the future.

For clarity, we provide a trivial example:

$$\texttt{ex:Monkey}\,(?x) \wedge \texttt{ex:hasUncle}\,(?x, ?y) \Rightarrow \texttt{ex:MonkeysUncle}\,(?y) \tag{2.1}$$

According to the above SWRL rule, which uses classes and predicates from a fictitious example "ex" vocabulary, if *?x* is an instance of the `ex:Monkey` class, and there is an assertion of the `ex:hasUncle` predicate that relates *?x* and *?y*, then *?y* is an instance of the `ex:MonkeysUncle` class.

2.1.1.5 SPARQL Query Language for RDF (SPARQL)

SPARQL [16] is a programming language, whose purpose is to express queries across RDF data sources, such as triple- and quad-stores.

The syntax for SPARQL is derived from both the SQL database query language and the Turtle RDF serialisation, and is designed to facilitate the description of RDF constructs, such as triples, in a serialisation-independent manner, i.e., by allowing users to specify the high-level structure and components of the required RDF triples, rather than their low-level serialisation-specific representation.

The most "basic" use case for SPARQL is the specification of Basic Graph Pattern (BGP) queries, where each BGP is a template for the subject, predicate and object of an RDF triple, whose components may be either hard-coded as RDF resources or literals, or bound at query-time[6] to free variables. For more complex use cases, the language includes constructs for the combination and manipulation of BGPs, including: logical conjunction and disjunction of BGPs; and, declaring a BGP as either required or optional.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE {
  ?agent
    foaf:name ?name ;
    foaf:mbox ?mbox .
}
```

Figure 2.2: Exemplar SPARQL query that uses terms from the FOAF vocabulary in order to select the name and mailbox ("mbox") of each agent that is described by the RDF data source.

An exemplar SPARQL query is given in Figure 2.2. The query uses terms from the FOAF vocabulary in order to select the name and mailbox ("mbox") of each agent that is described by the RDF data source. The query is an example of the "SELECT" form, which, when executed, returns the specified variables, and their bindings, directly. Finally, the query consists of two BGPs, which share a common subject.

2.1.2 Linked Data

Linked Data refers to a set of best practices for the dissemination of structured data on the Web [17].

In his original note [18], Berners-Lee outlines the four principles of Linked Data, which are intended to codify the expectations of a user agent for the behaviour of a software system that provides Linked Data. These principles are paraphrased as follows:

1. Use URIs to identify resources.
2. Use HTTP as the scheme, so that URIs may be dereferenced.

[6] The point in time at which a query is processed by the system.

3. When a URI is dereferenced, the system should respond with a machine-processable description of the identified resource, using standardised technologies, such as RDF and SPARQL.

4. Descriptions should include assertions of relationships to other resources, i.e., hyperlinks.

An application of the Linked Data principles has been demonstrated by the Linking Open Data (LOD) community project[7], which aims to publish and relate resources from a wide variety of open datasets, referred to collectively as the "LOD cloud". In September 2011, it was reported that the LOD cloud contained nearly 300 datasets, consisting of over 31 billion RDF triples [19].

2.1.3 Commonly-used Vocabularies

In this chapter, we have described knowledge representation technologies that facilitate the construction, manipulation and interrogation of machine-processable content on the Semantic Web. With this background, it is possible to use these technologies to describe conceptual entities that are relevant to scientific research. However, for many domains of discourse, the relevant conceptual entities have already been defined. Hence, we now proceed to describe popular schemas, ontologies and controlled vocabularies.

2.1.3.1 Dublin Core

The Dublin Core Metadata Initiative (DCMI) is a standards body, which focuses on the definition of specifications, vocabularies and best practice for the assertion of metadata. The DCMI has standardised an abstract model for the representation of metadata records [20], which is based on RDF and RDFS, and is composed of three sub-models:

The DCMI Resource Model – An abstract model, where resources are described by sets of assertions (property-value pairs). Each property is specified by a vocabulary, and each value is either a literal or a non-literal.

The DCMI Description Set Model – An extension of the resource model, whereby individual resources may have multiple descriptions, which form a set (referred to as a "description set").

The DCMI Vocabulary Model – An abstract model for the specification of vocabularies, i.e., sets of terms, where each term describes a class, property, vocabulary encoding scheme, and/or syntax encoding scheme. Using the vocabulary model,

[7] http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData

resources may be asserted as instances of a class, or as members of an encoding scheme.

The DCMI Metadata Terms (DCTERMS) is a specification of all metadata terms that are maintained by the DCMI, which includes: classes, properties, vocabulary encoding schemes, and syntax encoding schemes [21]. The specification incorporates, and builds upon, fifteen "classic" (legacy) metadata terms, which are defined by the Dublin Core Metadata Element Set [22].

Finally, the specification is encoded as an RDF schema[8].

2.1.3.2 Friend of a Friend (FOAF)

The purpose of the FOAF project is to facilitate the construction and dissemination of FOAF profile documents [23].

A FOAF profile document is a machine-processable information resource, which asserts metadata about people, organisations or other social groups, using the terms specified by the FOAF vocabulary[9]. It is intended (by the FOAF designers) that profile documents are managed by their owners, such that data is decentralised. The advantage of this approach is that it facilitates the aggregation of profile documents by directory services [24].

The metadata terms defined by the FOAF vocabulary are divided into three categories:

Core – Terms for describing the characteristics of people and social groups, which are independent of time and technology; used to describe basic information about people in present day, historical, and digital library contexts.

Social Web – Terms for describing user accounts, address books, and other Web-based activities, such as the use of online social networking and messaging systems.

Linked Data Utilities – Terms that facilitate integration with other Linked Data projects, including entity- and subject-oriented constructs, which may be used to indicate the "topic" or "focus" of a document.

2.1.3.3 Semantically-Interlinked Online Communities (SIOC)

The goal of the SIOC project is to specify "the main concepts and properties required to describe information from online communities," including, but not limited to, message boards, wikis and weblogs [25]. Hence, the purpose of SIOC is to provide a mechanism for integrating information that is exposed by disparate online communities.

[8] http://dublincore.org/2012/06/14/dcterms.rdf

[9] http://xmlns.com/foaf/spec/index.rdf

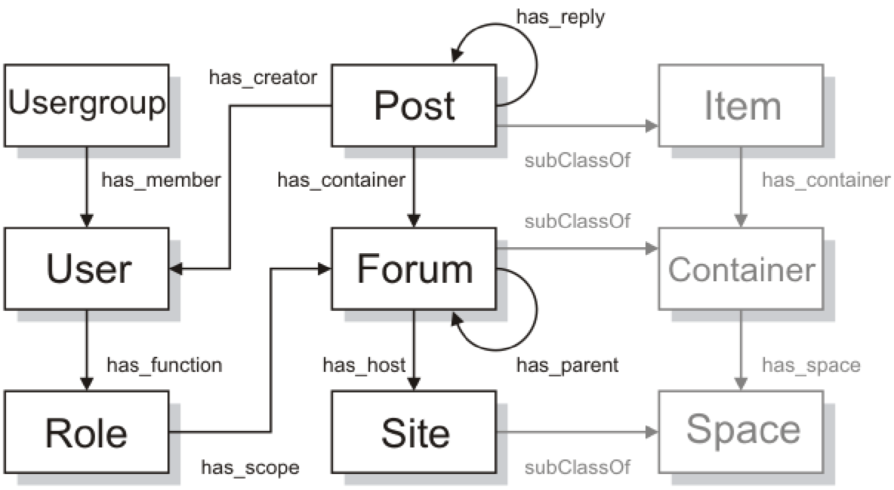


FIGURE 2.3: Depiction of the entity-relationship model for the SIOC Core Ontology (figure taken from http://www.w3.org/Submission/2007/SUBM-sioc-spec-20070612/).

The metadata terms of SIOC are specified as an OWL ontology, referred to as the SIOC Core Ontology[10] (depicted in Figure 2.3). The ontology is designed to be highly cohesive, i.e., metadata terms are only specified by SIOC if they are not provided by other schemas and ontologies. Hence, it is trivial to integrate other schemas and ontologies, such as DCTERMS, FOAF, etc. Moreover, the ontology is highly extensible, with a number of modules available via the SIOC project website[11].

In many ways, SIOC may be viewed as an extension of FOAF, providing an enhanced vocabulary for the description of the relationships that exist between users of online communities and their content, e.g., the ontology defines the `sioc:User` class, which is a subclass of `foaf:Person`; and, the `sioc:Post` class, which is a subclass of `foaf:Document`.

2.1.3.4 OAI Object Reuse and Exchange (OAI-ORE)

The goal of the OAI-ORE project is “to define standards for the description and exchange of aggregations of Web resources” [26].

The motivating example for the project is a record from the arXiv[12] repository for physics, mathematics, and computer science publications. In arXiv, records are assigned a URI, which resolves to a Web page (referred to as the “human start page”), containing basic bibliographic metadata, including: the title, date of creation, date of publication, forward and backward citations, etc. The human start page also links to alternate representations of the record, and to other records in the same arXiv collection.

[10] http://rdfs.org/sioc/ns
[11] http://sioc-project.org/
[12] http://arxiv.org/

OAI-ORE aims to address two issues, which are highlighted by the example: the identity of an aggregation, and the description of the constituents of an aggregation. First, the URI of the human start page is often used as the URI of the entire arXiv record. This is not appropriate, as, in the example, the URI identifies the human start page, and not the arXiv record. Hence, OAI-ORE provides a mechanism for the association of distinct URIs with both the aggregation itself and its aggregates (when appropriate). Second, as its name suggests, the human start page is both human-readable and human-processable, however, it is neither machine-readable nor machine-processable. Consequentially, the human start page cannot assert – in a machine-processable manner – the demarcation of resources, i.e., the constituents of an aggregation, and hence, the boundary between pairs of aggregations. Thus, OAI-ORE specifies a suite of formats for the representation of descriptions [27, 28, 29], along with a profile for the assertion of basic bibliographic metadata.

The OAI-ORE data model [30] defines four main conceptual entities:

Aggregation – An instance of the `ore:Aggregation` class, which denotes a set of aggregated resources.

Aggregated Resource – An instance of any class, which has been asserted to be a constituent of an aggregation, by the resource map that describes said aggregation.

Resource Map – An instance of the `ore:ResourceMap` class, which describes an aggregation.

Proxy – An instance of the `ore:Proxy` class, which denotes an aggregated resource that exists within the context of a specific aggregation.

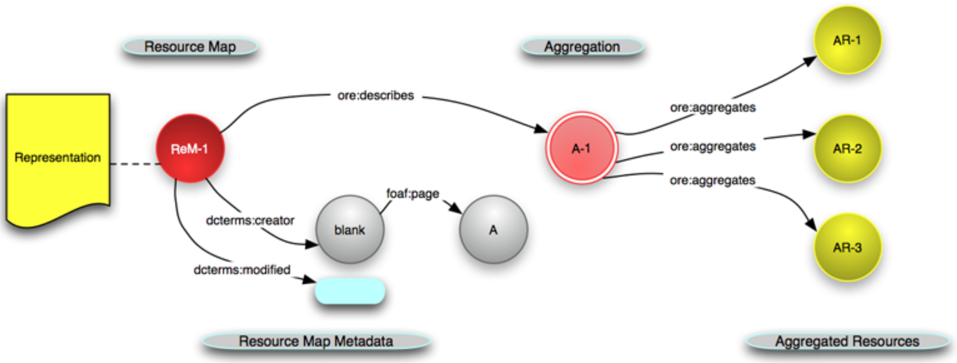


Figure 2.4: The aggregation A-1 aggregates three resources and is described by resource map ReM-1 (figure and caption taken from `http://www.openarchives.org/ore/1.0/primer.html`).

Interactions between the core conceptual entities are depicted in Figure 2.4, and are summarised as follows:

- The resource map "ReM-1" describes an aggregation "A-1" of three Web resources "AR-1", "AR-2" and "AR-3".
- The resource map is identified by a protocol-based URI, which may be resolved, yielding a machine-processable representation (of said resource map) "Representation".
- The resource map asserts basic bibliographic metadata, including: the creator (of said resource map), along with along with a hyperlink to the creator's home page "A"; and, the latest date of modification (for said resource map).

In isolation, there are two key drawbacks to the use of OAI-ORE: a non-prescriptive metadata and linking policy, and the absence of a non-repudiation strategy. First, while the OAI-ORE specification does prescribe how a resource map and aggregation should be constructed and annotated with metadata, it does not prescribe how to describe the constituent resources. The key implication of this approach is that it forces the "describer" to make arbitrary decisions. Consequentially, it is non-trivial to construct automated software systems that can recognise and manipulate the constituent resources. Second, it is not possible to establish the fixity of an OAI-ORE resource map, i.e., the specification does not specify a strategy for non-repudiation, e.g., the one-time calculation of a digital signature. The main implication of this approach is that there can be no distinction between "open" and "closed" resource maps.

To remedy these issues, Bechhofer, et al., introduce the abstract concept of a "Research Object (RO)" [31] – a semantically rich aggregation of information resources, which encapsulates a single "unit of knowledge", such as: a description of a recipe, a description of a scientific experiment, or a description of a medical procedure. When realised concretely, as metadata profiles, ROs provide a common foundation for the implementation of software systems.

2.1.3.5 Simple Knowledge Organisation System (SKOS)

The goal of the SKOS project is to enable the publication of controlled vocabularies on the Semantic Web, including, but not limited to, thesauri, taxonomies, and classification schemes [32]. However, it should be noted that, as controlled vocabularies do not formally assert axioms or facts, strictly speaking, SKOS is not a knowledge representation technology. Hence, as its name suggests, it is simply an organisation system, which relies on informal methods, such as the use of natural language.

The SKOS data model [33] is based on RDF and RDFS, and defines three main conceptual entities:

Concept – An instance of the `skos:Concept` class, which describes a single "unit of thought", e.g., a conceptual entity.

Concept Scheme – An instance of the `skos:ConceptScheme` class, which describes an aggregation of one or more SKOS concepts.

Collection – An instance of the `skos:Collection` or `skos:OrderedCollection` classes, which describes a labelled and/or ordered group of SKOS concepts.

In SKOS, a concept scheme may contain descriptions of many concepts. Moreover, often, said concepts do not exist in isolation, but instead, are related to one another by meaningful links (referred to as "semantic relations"). The SKOS data model distinguishes between two types of semantic relation: hierarchical and associative. A hierarchical link between two concepts indicates that the domain is more general ("broader") than the codomain ("narrower"). An associative link between two concepts indicates that the domain and codomain are "related" to each other, but not by the concept of generality.

SKOS provides a basic vocabulary of metadata terms, which are used to associate lexical labels with resources (of any type). Specifically, SKOS allows consumers to distinguish between the preferred, alternative and "hidden" lexical labels for a given resource. As their names suggest, the preferred and alternate lexical labels are amenable to inclusion in human-readable representations. Moreover, the "hidden" lexical labels for a given resource are particularly useful when developing systems that rely on text-based queries to locate resources, e.g., common mis-spellings may be associated with a resource, to enable its subsequent discovery, without encouraging further spelling mistakes.

2.1.3.6 Vocabulary of Interlinked Datasets (VoID)

The purpose of the VoID project is to specify a vocabulary for the description of RDF datasets [34]. The motivation for VoID is to provide a bridge between the producers and consumers of Linked Data, i.e., to facilitate automated dataset discovery, and to enable the curation and archival of datasets.

The specification for the VoID vocabulary defines four types of metadata:

General metadata – Includes basic bibliographic metadata, such as the title, description and license for the dataset, using terms that are defined by Dublin Core.

Access metadata – Terms for asserting the methods by which the RDF triples that comprise a dataset may be accessed, including: the textual format of resolvable URIs, and the location of SPARQL end-points.

Structural metadata – Terms for asserting the high-level schema and internal structure of an RDF dataset. This may include the vocabularies that have been used in the dataset, statistics about the size of the dataset, and examples of prototypical resources.

Description of links between datasets – A link-set is an instance of the `void:Linkset` class (a subclass of `void:Dataset`), which describes the relationship between two RDF datasets. The motivation for defining a conceptual entity to denote a linkset is to facilitate navigation between RDF datasets.

A prominent use of VoID is the Linking Open Data (LOD) cloud diagram [19], which is procedurally generated from a collection of VoID data- and link-set descriptions. The diagram depicts the relative size of each RDF dataset, along with its relationships to other datasets.

2.2 Laboratory Notebooks

To conduct research is to enact the scientific method – a cyclic methodology for the acquisition of knowledge. First, a question is formulated. Second, both a falsifiable and a null hypothesis are conjectured, and their logical implications are explored. This is followed by the planning and enactment of a controlled experiment, whose results are analysed, given the context of the two hypotheses. Finally, the original question is answered, new questions are formulated, and the cycle is repeated.

An outcome of the enactment of the scientific method is the generation of content (data, information, and, hopefully, knowledge). Hence, an obvious, and reasonable, question to ask is: where does this content reside? The answer, somewhat unsurprisingly (given the title of this section), is inside of a "laboratory notebook" – an artefact, whose primary function is to persist and manage the content that is generated during the enactment of the scientific method, i.e., to provide a record of the activities of one or more researchers.

For many researchers, the value-proposition for using a laboratory notebook is as personal, and as diverse, as the content that is being persisted. However, ostensibly, their rationale is informed by two key motivations. First, and foremost, the use of a laboratory notebook is driven by the human need for cognitive delegation, i.e., by delegating the persistence and retrieval of content to their laboratory notebook(s), the cognitive resources of the researcher are freed, and made available for use in other endeavours. For example, the eminent, English scientist Michael Faraday maintained an extensive collection of laboratory notebooks because he was "mistrustful of his own memory" [35]. Second, the use a laboratory notebook necessarily increases the potential for the realisation of ephemeral value at indeterminate points in the future, i.e., after it has been persisted, the content of a laboratory notebook may be repurposed and reused.

Clearly, the above motivations are generic, and applicable not only to the use of laboratory notebooks, but also to the use of any Content Management System (CMS)—software systems, whose capabilities include the management of content. Thus, in order to understand the value-proposition for using a laboratory notebook specifically, it is necessary that we distinguish between the distinct value-positions for the use of a generic CMS, and for the use of a CMS that has been specialised for one or more domains of discourse. Furthermore, we must posit how the act of specialisation affects both the degree of utility that is afforded by cognitive delegation, and the effect (if any) that this has on the potential for the realisation of ephemeral value.

For completeness, we now list some (of the many) benefits of using a CMS:

Dissemination – Content that has been persisted using a CMS may be retrieved at any time in the future, facilitating its repurposing and reuse. However, it is important to note that absolute fidelity can only be provided if the CMS ensures that, after it has been persisted, content is never modified. We note that, for paper-based laboratory notebooks, this matter is a discipline on the part of the researcher.

Identity – In order to facilitate retrieval, it is necessary for the CMS to assign one or more identifiers to each unit of content, which may subsequently be referenced and resolved. For paper-based laboratory notebooks, these identifiers may be relative, such as "the graph on page 10 [of a specific paper-based laboratory notebook]", or absolute, such as "the entry for May 5th 2012 [by a specific researcher]".

Metadata – Identifiers may be referenced as the subject or object of logical assertions, including: structural or semantic constraints; bibliographic annotations, such as the date of creation, or the list of contributing authors; and, provenance information, such as the date of the most recent modification.

Versioning – As stated earlier, content that has been persisted using a CMS should never be modified. Instead, a completely new version should be created, with a reference to the previous version. Moreover, for readers to be able to trust that content has not been modified, the CMS should provide ample provenance information.

State – Within the context of a CMS, content is typically managed according to one or more state machines (referred to as "life-cycles"), where each machine is always in exactly one state at any point in time (referred to as the "current state"). Moreover, machines may transition between states, if predetermined conditions are met. Hence, within the context of a CMS, the ability to perform certain actions, such as the persistence of a new version of a unit of content, or the retrieval of a specific unit of content, may be restricted, given the additional context of the current state. Furthermore, if it is permitted, the act of transitioning between states can be remotely witnessed by a third-party.

Aggregation – Content that has been persisted using a CMS may be grouped together, explicitly delineated, and referenced as a distinct logical unit with its own identity and metadata (referred to as an "aggregation"). Moreover, the CMS as a whole may be regarded as an implicit aggregation of the sum of its content. However, without semantics, the aggregations themselves are purely structured delineations (of other content), and hence, posses no additional qualities, e.g., without semantics, an aggregation does not "correspond to" any other concept.

Security – Content that has been persisted using a CMS may be subject to an access control policy, whereby only authorised users, who have authenticated with the system, are granted specific permissions and capabilities. As we have alluded to, access control policies may also be informed by the current state of the content, e.g., a new version of a unit of content cannot be created unless said content is in the "draft" state; or, a unit of content cannot be viewed by third parties unless it is in the "published" state.

Protection of Intellectual Property – As we have explained, within the context of a CMS, each unit of content is assigned an identity, described by metadata (including provenance information), and secured by an access control system. When used in combination, these capabilities enhance the ability of the researcher to secure intellectual property rights for their scholarly works.

As we have shown, many benefits can be derived from the use of a generic CMS. However, it is important to note that none of these benefits can be attributed, in any way, to the nature or specific qualities of the content that is being persisted, i.e., on the whole, the functionality of a CMS is generic, and agnostic to its content. Instead, we must conclude that any benefits that are derived from the use of a CMS, which has been specialised for one or more domains of discourse, must be attributed to the act of specialisation itself.

Consequentially, we argue that the value-proposition for the use of a CMS, which is specialised for one or more domains of discourse, is actually a combination of three distinct value-propositions, which must first be considered separately, and then together as a set:

1. The value-proposition for the use of a generic CMS;

2. The value-proposition for the use of a nomenclature that is specific to one or more domains of discourse; and,

3. The value-proposition for the integration of (2) with (1), i.e., the value-proposition for the act of specialisation.

Accordingly, we now introduce the concept of a nomenclature, and describe the impact of its incorporation into a generic CMS.

2.2.1 Nomenclature

A nomenclature is a formal system for naming things; a morphism, from the domain of things, to the codomain of names.

Ostensibly, and somewhat obviously, the purpose of a nomenclature is to assign names to things, i.e., given a thing as input, a nomenclature allows us to generate a name as output. Hence, given the context of a specific nomenclature, two or more parties may converse with each other, and share information, where the nomenclature is used as a *lingua franca* (or common language).

However, less obviously, nomenclature also has another, more subtle, purpose. If two or more parties explicitly agree to use the same nomenclature, then they also implicitly agree on the existence and nature of three mathematical entities:

- A set of things (the domain);
- A set of names (the codomain); and
- A formal process for the consideration of a subset of the aspects of each thing, and the subsequent assignment of one or more names (the morphism).

Furthermore, given the existence of the above mathematical entities, the two parties implicitly agree that if multiple things are assigned the same names, then said things are equivalent to each other, with respect to the subset of aspects that are being considered, i.e., given the context of a specific nomenclature, the predicate that relates a thing to a name is inverse-functional. Moreover, the two parties implicitly agree that the opposite is also true, i.e., given the context of a specific nomenclature, if multiple things are assigned different names, then said things are disjoint to each other, with respect to the subset of aspects that are being considered. Thus, given the context of a specific nomenclature, any name may be used for discrimination purposes.

However, it is important to note that, in this context at least, discrimination is not equivalent to resolution. Generally speaking, given a name, while it is possible for two parties to agree that they are "using the same name", it is not possible to locate "the thing [or set of things] with a given name". This is for two key reasons. First, it assumes that the morphism from things to names is injective, i.e., a one-to-one mapping. Second, it assumes that the things themselves do not permit a canonical representation. If either of these assumptions are invalid, then the morphism is necessarily surjective, i.e., a many-to-one mapping, and hence, non-invertible. Thus, in order to confer the quality of "resolvability" to a set of names, we must specify an arbitrary formal process (referred to as a "resolution scheme" or "resolution protocol").

As we have discussed, the agreement between two parties to use the same nomenclature is motivated by the shared functional requirement for domain-specific data integration capabilities. However, as we have noted, the terms of these agreements concern the use of a nomenclature, and not the use of a CMS, i.e., it is the nomenclature that affords data integration capabilities to the CMS, and not *vice versa.* Thus, the incorporation of a specific nomenclature into a generic CMS affords said CMS a specialisation for a specific domain of discourse. Therefore, the content of any two CMSs, which share a subset of nomenclature(s), and hence, are specialised for the same domains of discourse, may be integrated.

2.2.1.1 Domain-specific Nomenclature

At this point, we have introduced the concept of, and described the purpose of, a nomenclature, but, we have not presented any specific examples. Moreover, we have deliberately maintained a very high level of abstraction. The key reason for this is that, in order to understand the characteristics of a domain-specific nomenclature (in general), and not of a specific example of a domain-specific nomenclature, we must avoid restricting our considerations to a specific domain of discourse. However, since a nomenclature is a system for the assignment of names to things, but some things are specific to one or more domains of discourse, then in order to proceed further, we must answer the following questions:

- Is the set of "nameable" things specific to a given domain of discourse?
- Within the context of nomenclature, which entity (or entities) confer the quality of "domain-specificity" to a given domain of discourse?

To answer the first question, we observe that, since any thing that does not have a name can be referred to temporarily using a pronoun, all things that are conceivable, are, in principle, "nameable". For example, if "it does not have a name", then, in fact, it does have a name. In this case, the pronoun "it". Furthermore, we note that, since anyone may [conceivably] conceive of any [conceivable] thing, the set of "nameable" things is isomorphic to the set of conceivable things. Thus, we infer that the set of "nameable" things is not unique to a specific domain of discourse. Moreover, we conclude that it is not the set of "nameable" things that confers "domain-specificity" on a given nomenclature.

Given the above argument, there remain two candidates for the role of conferring the quality of "domain-specificity": the set of names (for things), and the formal processes by which said names are assigned to "nameable" things. Clearly, the names themselves are not specific to a given domain of discourse, as each name is simply a unit of information, whose structure is interpreted according to a given semantics. Therefore, we infer that it must be the formal processes for the assignment of names to "nameable" things, and the semantics for said names, that are domain-specific.

In conclusion, the decision to use domain-specific nomenclature is motivated by the need for data integration capabilities. Content that features domain-specific nomenclature is conferred the quality of "domain-specificity", and the names themselves are conferred their own structure and semantics. However, as we have noted, just as it is possible for anyone to conceive of any conceivable thing, it is also possible for any content to include any nomenclature. Therefore, we must conclude that, within the context of a CMS, the act of specialisation for one or more domains of discourse is in fact a generic operation, given one or more specific nomenclatures.

2.2.2 Paper-based Laboratory Notebooks

A paper-based laboratory notebook is a laboratory notebook that has been constructed by binding together one or more sheets of paper (referred to as "pages").

Typically, the pages of a paper-based laboratory notebook serve to physically delineate its content, i.e., a new page is started for each unit of research. Moreover, it is common for the pages of a paper-based laboratory notebook to be assigned an implicit chronological ordering, in correspondence with the causality of the research that is being described therein. Hence, an important consideration about the use of a paper-based laboratory notebook is that, for the provenance of its content to be legally acceptable, the ordering of the pages must remain fixed, i.e., if an old page is torn out, or a new page is sewn in, then the provenance of the content becomes inconsistent, and thus, is no longer legally acceptable.

Of course, such considerations of consistency apply only to the ordering of the pages of a paper-based laboratory notebook, and not to the content that is described therein. Hence, it is relevant to consider, within the context of the content of a paper-based laboratory notebook, the difference between consistency and correctness. Put simply, the content of a paper-based laboratory notebook is always consistent, and sometimes correct. Said differently, in a paper-based laboratory notebook, we may consistently assert incorrect information, e.g., it is possible to write $2 + 2 = 5$ on the surface of a piece of paper, without causing said piece of paper to become inconsistent[13]. In contrast, within the context of a formal system, such as a software application, it is impossible to assert inconsistent information, as doing so would be an "error".

When the logical implications of the above statement are fully considered, it becomes clear that part of the utility of a paper-based laboratory notebook is derived from a characteristic of the underlying medium: the physical information that describes a piece

[13] At time of writing, humanity has been unsuccessful in constructing an artefact with inconsistent physical information. In fact, many scholars speculate that it would be impossible to do so. However, with tongue in cheek, we would like to posit that, in the unlikely event that such an artefact is constructed, instead of triggering a Universe-ending paradox, said artefact would spontaneously combust, returning the system to a consistent state. Unfortunately, this does mean that researchers will be forced to bear witness to their work bursting into flames. However, at least they will live to tell the tale!

of paper is disjoint to the information content that is encoded on the surface of said piece of paper. On its own, a piece of paper has no semantics. Hence, the state of a piece of paper, and by extension, anything that is constructed from said piece of paper, is always consistent. In contrast, the content of a piece of paper has semantics. Thus, the correctness of the state of the content of a piece of paper can only be determined retrospectively, given said semantics.

2.2.3 Electronic Laboratory Notebooks

An Electronic Laboratory Notebook (ELN) is a software system, whose components, when used in combination, implement some or all of the functional requirements of a paper-based laboratory notebook. Hence, an ELN may be regarded as a digital emulation of a paper-based laboratory notebook, where one or more of the components have been specialised for a specific domain of discourse. To characterise an ELN, we consider the following criterion:

Paper Use – The amount of paper that is consumed;

Incorporation of Structure – Whether or not the content of ELN entries is represented as unstructured text or as structured objects; and

Incorporation of Semantics – Whether or not the content of ELN entries is given machine-processable semantics.

Clearly, the amount of paper that may or may not be consumed when using a specific ELN is variable. Hence, for simplicity, we define two broad categories of ELN: paperless and hybrid. In a paperless ELN, as the name suggests, the use of paper is minimised or avoided completely, and software components are used for all aspects of data capture and reuse. By contrast, in hybrid ELNs, researchers must generate separate, paper-based counterparts for each digital information resource, which are subsequently managed by one or more software components. Hence, hybrid ELNs may be further categorised, according to whether or not said paper-based counterparts are transient (after ingest) or persistent.

2.2.3.1 Characterisation of ELN Content

Ostensibly, the purpose of an ELN is identical to that of a paper-based laboratory notebook: to persist the content of its entries. However, as Elliott [36] states, a core functional requirement, from the perspective of end-users, should be “integrating an ELN with other systems in the enterprise, most notably LIMS, document management, instrument data systems, data archiving and/or scientific databases.” Hence, to characterise a specific ELN, it is also necessary to characterise the content of its entries.

For our characterisation, we assume that an entry in an ELN is analogous to a box, which may contain an arbitrary amount of content. We have found this analogy to be particularly apt, as it facilitates the separate consideration of the nature of boxes, and of the nature of the content of boxes.

Some relevant capabilities of boxes are as follows:

- The capability to be opened by one or more specific individuals;
- The capability to have its contents modified by one or more specific individuals;
- The capability for the exterior of the box to appear as either transparent or opaque, when observed by specific individuals;
- The capability to be sealed by one or more specific individuals; and
- The capability to restrict its contents according to specific criterion.

From a software engineering perspective, each capability clearly corresponds to the implementation of one or more aspects of a generic software system, e.g., the capability for a box to be opened and modified corresponds to the implementation of an access control system; the capability to be sealed corresponds to the implementation of an electronic signature system; and, the capability to restrict contents corresponds to the implementation of generic class taxonomies. Moreover, it is immediately obvious that a capable "box management system" should have other capabilities, e.g., the capability to record the application of other capabilities.

	No Structure	**Structure**
No Semantics	Plain text	Markup
Semantics	Plain text with domain-specific entities	Objects

Table 2.1: Characterisation of the content of ELN entries, given the presence or absence of structure and semantics.

In Table 2.1, we give our characterisation of the content of ELN entries. In the absence of both structure and semantics, content is both persisted and presented as human-readable plain text. If structure is defined, then content is persisted in a machine-processable representation, and is presented, via one or more transformations, as human-readable, formatted markup. In contrast, if semantics are defined, then it may be assumed that plain text contains references to domain-specific entities and nomenclature, which may subsequently be extracted via the use of a deterministic automaton. Finally, in the presence of both structure and semantics, content is both persisted and presented in a machine-processable representation.

2.2.3.2 Critique

We now proceed with our critique of exemplar ELN implementations, and supporting software platforms.

Collaboratory for Multi-scale Chemical Science (CMCS) The Collaboratory[14] for Multi-scale Chemical Science (CMCS)[15] is a software architecture and informatics portal toolkit, whose goal is to facilitate multi-scale collaboration between individual research groups and larger communities, in the domain of combustion science. Combustion research was selected as it relies on the integration of chemical information and data spanning more than nine orders of magnitude (in terms of both the length and timescale dimensions). Myers, et al., [37] argue that "the major bottleneck in multi-scale research today is in the passing of information from one level to the next in a consistent, validated and timely manner." For multi-scale research, the issue of data integration is particularly irksome, as, generally, data is heterogeneous, being represented using a wide variety of conceptual models and formats. Hence, the challenge for the CMCS developers was to develop a generic, multi-scale informatics portal toolkit, whilst integrating support for both domain- and scale-specific models, formats, and software applications.

To address the issue of data integration, the CMCS developers adopted a two-fold strategy of an aspect-oriented design, which focused on the use of open-source technologies. First, an aspect-oriented design was used for the software architecture, whereby, instead of standardising a common conceptual model, which would be reused through-out, the developers opted to individually specify the unique aspects of each integration point. The key benefit of this approach, is that it necessarily requires that "aspects" are reified, such that they may be explicitly identified. Hence, by adopting an aspect-oriented design, the developers of CMCS were able to decouple the implementation of the software architecture from its subsequent usage, i.e., assuming that the specification for an integration point remains invariant, the domain-specific conceptual models, vocabularies and data formats may continue to evolve, without affecting the implementation of the software architecture. Second, wherever possible, in all areas of the implementation of the software architecture, the CMCS developers leveraged standard technologies, only developing their own solutions when open-source or *de facto* alternatives were unavailable. The key benefit of this approach is that, from a software engineering perspective, the resulting software architecture is relatively light-weight, with fewer dependencies on proprietary code, and hence, is more extensible, and amenable to future modification.

In CMCS, the tasks of data management and integration are both delegated to a special-purpose software framework – Scientific Annotation Middleware (SAM) [38]. By leveraging SAM, CMCS implements a paperless ELN, whose user interface is provided by

[14] The term "collaboratory" is a contraction of "collaborative laboratory".

[15] http://cmcs.org/

a Web-based portal [39]. The key advantage of this approach is that, by delegating generic tasks to SAM, the CMCS developers were able to focus their collective energies on domain-specific tasks [40], such as the standardisation of community-specific vocabularies.

Prism Tabard, et al., have presented Prism, a hybrid-persistent ELN, whose primary purpose is to provide a coherent view of the multiple, parallel sequences of events, which manifest during the day-to-day activities of Bioinformatics researchers [41]. The developers of Prism opted for a sequence manipulation paradigm, where the elements of each sequence (referred to as an "activity stream") are descriptions of events that have transpired in the past. There are two key benefits to this approach. First, and foremost, from a software engineering perspective, sequences are well-studied data structures, for which a significant corpus is available. Second, rigid adherence to a single paradigm results in a conceptually simple design, allowing users to reason consistently about the expected consequences of their actions. For example, Prism allows users to create new activity streams (from pre-existing ones), by levering the standard operations of sequence filtering, concatenation and sorting. Moreover, individual activity streams may be shared between users.

In principle, the fundamental disadvantage to the Prism approach is that, as it is hybrid-persistent, it requires that, for each ELN entry, users maintain a separate counterpart in their pre-existing, paper-based laboratory notebook. Hence, the deployment and use of Prism actually increases the level of data redundancy in the system. From a data curation perspective, this could be viewed as a benefit, as, when annotated and linked together, multiple representations of the same information may provide additional context. However, to the users of the software, the requirement to designate and maintain a "master notebook", to which all other notebooks are synchronised, incurs a significant cognitive overhead, and may, as a result, give rise to inconsistent data. This issue is observed by the developers, who note that, during the evaluation period, "when given the choice, people preferred to use the paper format as the master notebook," indicating that the value-proposition for using a hybrid-persistent ELN, such as Prism, may not be justifiable, even to computer-literate users, such as Bioinformaticians, whose day-to-day research is primarily conducted *in silico*.

Comb*e*Chem Comb*e*Chem[16] is an inter-disciplinary research project at the University of Southampton. The goal of Comb*e*Chem is to demonstrate a compelling value proposition for the use of Semantic Web technologies as part of Chemistry research methodologies, in particular RDF, and to enable the publication and dissemination of machine-processable descriptions of scientific experiments and their outcomes. To this end, the core principles of Comb*e*Chem are the capture of semantic annotations "at

[16] http://www.combechem.org/

source" [42], and the community-driven development of machine-processable representations of domain-specific nomenclature.

As part of the Comb*e*Chem project, a methodology was developed for the collaborative design and implementation of descriptions of formal processes – Making Tea [43]. To test the methodology, an inter-disciplinary group of computer scientists and domain specialists attempted to model the "safe, repeatable procedure" of making a cup of tea using both household and laboratory equipment. In the evaluation of their approach, schraefel, et al., note that "Making Tea did not make us domain experts". To the contrary, the key benefit of the Making Tea approach is that it opens a direct channel of communication between software engineers and domain specialists, facilitating the development of a *lingua franca*, and allowing both parties to focus on their individual goals.

The Comb*e*Chem software architecture is a composition of three core components: a paperless ELN – Smart Tea [44]; a generic data management and integration framework; and, a software application for *a priori* process modelling of scientific experiments (referred to in publications as the "Planner" or "planning tool"). The software components are informed by three ontologies: a vocabulary for basic chemical information, a vocabulary for physical quantities, measurements and units of measure; and, a vocabulary for process modelling. Taylor, et al., [45] note that "it would be impossible to predict in advance the way that data would be accessed and used," hence, the vocabularies were designed for maximum extensibility, rather than expressiveness. The advantage of this approach is that, by provisioning for generic use cases, the Comb*e*Chem software architecture may be specialised for non-Chemistry domains, whilst maintaining the ability to integrate heterogeneous data. However, the disadvantage of the Comb*e*Chem approach is that, by developing their own custom solutions, the Comb*e*Chem developers risk reinventing the square wheel, by overlooking prior art, and re-engineering solutions that are of a lower quality than the standard, e.g., Taylor, et al., acknowledge that it would be a significant effort "to build up the chemical ontology to the level comparable with … CML" [45].

Before a scientific experiment can be performed, the plan for said experiment must be assessed, so that any potential hazards can be identified, and the associated risks can be mitigated. In the UK, this is conducted as part of the Control of Substances Hazardous to Health (COSHH) assessment [46]. Accordingly, the layout for the user interface of the Comb*e*Chem planning tool is designed to emulate the elements of a compliant COSHH form. However, it should be noted that the apparent correspondence is only skin deep, as plans for scientific experiments cannot be integrated with occupational safety and health systems.

Ostensibly, there is one key reason for the limited data integration capabilities that are offered by Comb*e*Chem. Simply, the IUPAC InChI [47], which is suitable for discrimination purposes only, is misused as a nomenclature. This is a mistake. The purpose of an InChI is to provide a textual representation for small organic molecules. Hence, there are many varieties of chemical substances that cannot be identified using an InChI, including: polymers, mixtures, complex organometallics, excited state and spin isomers, etc. Thus, any substance that cannot be represented by an InChI cannot be referenced by Comb*e*Chem. The correct approach, is to coreference chemical substances using as many inverse-functional identifiers as possible.

eCAT Goddard, et al., present eCAT [48], a paperless ELN, whose user interface is entirely Web based, facilitating its use on a wide variety of Web-enabled devices. In eCAT, content is persisted as raw HTML, which can be edited using a WYSIWYG user interface. Interestingly, eCAT includes a basic meta-modelling environment, which allows users to specify domain-specific entity-relationship models, where each entity is a struct of typed attributes. The key benefit of this approach is that, by provisioning for generic use cases, eCAT may be specialised for researchers from any domain, as long as said researchers are willing to specify their own models. A further benefit of this approach is that, the act of specialisation necessarily creates a channel of communication between researchers, facilitating the eventual development of a *lingua franca*.

2.2.4 Summary

Given our analysis, we draw the following conclusions:

1. The overwhelming majority of the functional requirements for an ELN are generic.

2. To specialise an ELN requires open communication with domain experts, and the development of a *lingua franca*.

3. The most important functional requirement for an ELN is the capability to perform advanced data integration.

4. The fundamental difference between a paper-based laboratory notebook and an ELN is that, in a paper-based laboratory notebook, the information content of an entry is disjoint to the information that describes the content of said entry, i.e., the information in a paper-based laboratory notebook is “on” the paper, rather than “of” the paper. Whereas, in an ELN, the information content of an entry is (the same as) that which describes said entry. Hence, the state of a paper-based laboratory notebook is always consistent and correct, regardless of the consistency or correctness of the information content of its entries. Unfortunately, for many ELN implementations, consistency and correctness have become confused, and incorrect

information cannot be represented consistently. The result of this confusion is the unintended manifestation of limitations and restrictions to the software components and nomenclature that are provided to end-users. Therefore, to provide a true emulation of the functionality of a paper-based laboratory notebook, both the developers and end-users of an ELN must admit their fallibility, and permit the consistent representation of any and all information, regardless of its correctness, which, as we know, may always be determined retrospectively.

2.3 Applications of Knowledge Representation Techniques and Technologies for Research

In recent years, advances in the processing speed, memory capacity, and information bandwidth of computational technologies, coupled with an increasing demand from industry for the rapid generation of new products and services, has resulted in the convergence of computational technologies and the scientific method. An important outcome of this convergence, is the generation, use, and management of significant amounts of data, information, and knowledge [49].

This “data deluge” (as it is now being referred) is the result of the integration of high-throughput, on-line, and real-time data sources with software systems. These data sources include: sensor networks, parallel computation platforms, complex mathematical models, and other instrumented formal processes, such as instrument, apparatus and chemical sample monitoring systems.

For example, in the Pharmaceutical industry, the research, development, and manufacture of new drug products involves the integration of data from a diverse range of sources, including: both raw and derived experimental data, laboratory reports, and any other data that was used as part of a decision-making process [50]. In fact, the United States Food and Drug Administration (USFDA) requires that, as part of its initiative to adopt the principle of “Quality by Design (QbD)” [51], all data relating to the research, development, and manufacture of a new drug product must be provided as documentation, facilitating the verification and validation of published results, and the certification of the data’s integrity.

The transition towards data-intensive science also creates many new opportunities for both software developers and end-users. First, and foremost, the transition provides the opportunity for the development and implementation of new software systems, spawning numerous academic research projects, which, in turn, creates jobs for software developers. However, as Hey and Trefethen note, with the approaching data deluge, “the issue of how we handle this vast outpouring of scientific data becomes of paramount importance” [52]. They point out that the current approach of manual examination is not appropriate for the vast quantities of complex data that will be generated in the future, and, instead, argue that data should be automatically annotated with appropriate metadata at source (or at least, at the earliest available opportunity); a sentiment that is echoed by other authors [53, 54].

Clearly, there is an opportunity for both the software developers and the end-users to come together, and to form a community, whose goal should be the harmonisation of the nomenclature that will be used for data annotation. Given the availability of, and investment in, such nomenclature, communities will have the profound opportunity to regulate the data curation, archival, and preservation policies for their software systems [55].

Another benefit of the transition towards data-intensive science is the opportunity for research organisations to leverage the general increase in the availability and quality of computational resources, in order to automate and parallelise the planning and enactment of both physical and digital formal processes, e.g., to migrate away from a reliance on empirical experimentation, towards the "brute-force" computation of multivariate models and simulations (referred to as *in silico* experimentation), i.e., to "discover" phenomena, rather than to "explain" them, given the context of a mathematical framework. However, as Zhao, et. al., point out, automation through *in silico* experimentation is not, on its own, a panacea. Instead, they argue that "it is also vital to be able to understand and interpret the outputs of those experiments" [56], and argue that software systems must provide a description of the provenance of each experiment, so that the outcomes can be trusted, and the results can be replicated, by other researchers.

The remainder of this section is organised as follows. First, we introduce the concept of a computational workflow. This is followed by a discussion and critique of conceptual frameworks and controlled vocabularies for the exposition of the provenance of formal processes. Finally, conclusions are drawn.

2.3.1 Computational Workflows

As defined by the Workflow Management Coalition [57], a workflow is a description of a formal process, which will, when enacted, manipulate a given space of information resources, with the goal of achieving one or more desired outcomes. Workflow technologies are derived from service-oriented architectures [58]; an architectural paradigm for the development of software systems, which focuses on the composition of loosely coupled, distributed software components.

In a service-oriented architecture, the complexity of individual software components can vary greatly; ranging from the relative simplicity of the client-side execution of a command-line application, to the intricacies of communication with a "full stack" Web service. A key benefit of adherence to a service-oriented architecture is that, by encapsulating each unit of functionality inside of a distinct software component, where the semantics of said component are well-defined, the software system (as a whole) is afforded a greater degree of flexibility and scalability, e.g., in a service-oriented architecture, any software component may be substituted, at any time, without affecting the macro-scale functionality of the system, if and only if the replacement has an equivalent semantics. In fact, in the Life Sciences community, digital repositories have been developed specifically for the dissemination of both individual components [59] and whole workflows [60], facilitating discovery and reuse [61]. However, while individual software components may provide trivial solutions for trivial problems, the complexity of the situation increases sharply when a suite of services must be coordinated, in order to solve more complex problems. It is this coordination that is often provided by workflow technologies.

Recently, the concept of a workflow has been applied to data-intensive science, coining the term "scientific workflow" [62]. Scientific workflow systems are characterised by their adoption of simple computational models, in particular, the dataflow programming paradigm [63], where computations are described by directed graphs, whose vertices correspond to operations (on data), and whose edges correspond to the transportation of data between operations. Operations are modelled as "black boxes", with explicitly defined inputs and outputs, which may be evaluated as soon as all of their inputs are available. Hence, in scientific workflow systems, the order of execution (of individual software components) is determined implicitly by the "flow" of data through the workflow. This is in contrast to business workflow systems, where both the order of the execution of individual operations, and the strategies for the ordering of the execution of multiple operations, may be specified explicitly.

Given a survey of the literature [64, 65], we proceed to describe the following key functional requirements for scientific workflow systems:

Context-specific Abstraction – Ostensibly, the primary users of a scientific workflow system are researchers; domain specialists, who do not necessarily have expertise in software development. However, science is not a solitary activity, and researchers may be supported by many other individuals, some of whom may be expert software developers. Therefore, as scientific research is (most often) grounded in the planning and enactment of scientific experiments, and not in the low-level implementation of software systems, a scientific workflow system should provide a contextual user interface, whereby only relevant information, which is necessary for an understanding of the task, is presented to the current user, given their personal goals, at an appropriate level of abstraction, given their expertise, e.g., a lay user of the system, such as a researcher, whose primary goal is the representation of a methodology, should only be presented with domain-specific information, such as the purported purpose of each operation, and the semantics of each input and output.

Micro- and Macro-scale Composability – One of the most important qualities of scientific workflows is that they are composable. Like the individual services from which they are themselves composed, scientific workflows may be modelled as "black boxes", where the entire functionality of a workflow is encapsulated, and exposed as a single set of inputs and outputs [66]. A key benefit of this approach is that the process of decomposing a single, complex workflow into multiple, simple workflows naturally gives rise to a more cohesive and modular design, which, consequentially, is easier to extend and maintain. Moreover, the ability to "divide and conquer" the task of modelling complex processes encourages scientific collaboration [67].

Referential Transparency and Data Immutability – Inspired by the functional programming paradigm [68], the data model that is used by a scientific workflow system should, in the absence of side-effects, have the properties of both referential transparency and data immutability. The former states that, for a given set of inputs, the evaluation of an operation will always yield the same set of outputs. The latter states that the value of a data item (such as an input or output) may be initialised exactly once, and is subsequently immutable, i.e., may be read, but not modified. There are four key advantages to this approach. First, as both the operations and the coordination of said operations are referentially transparent, any scientific workflow that references an operation is automatically referentially transparent. Second, by assuming referential transparency, the scientific workflow system may infer an optimal evaluation strategy, e.g., by deferring computation until it is needed (referred to as "lazy" or "demand-driven" evaluation), or by automatically unrolling and parallelising iterative programming constructs. Third, the result of the evaluation of a referentially transparent operation may be memoized for latter reuse (at the cost of memory). Finally, as a direct consequence of the property of data immutability, a directed graph that describes the evaluation of a scientific workflow is necessarily acyclic.

Advanced Data Integration and Management – Not all data are created equal. Scientific software systems typically require access to heterogenous information resources, some of which may also be distributed [69]. Consequentially, there exists a core functional requirement for advanced data integration capabilities in scientific software systems, whereby multiple information resources can be integrated (homogenised) via a data mediation model. Within the context of scientific workflow systems, since the structure (inputs and outputs) and semantics (purported purpose) of each service, and hence, the workflow as a whole, are well-defined, it is reasonable to require that a scientific workflow system be capable of automatically selecting the appropriate mediation strategy (referred to as an "adapter" or "shim") for each unit of data [70]. Moreover, not all data are located physically, i.e., on a local disk. Hence, there is a core requirement for scientific workflow systems to be capable of staging data both in and out of the evaluation environment, where said staging is transparent to end-users of the software system. However, as Deelman and Chervenek note, not all data can be transported efficiently around a network [71], e.g., because it is simply too large, or because the latency and/or bandwidth of said network are not suitable. Therefore, in order for the scheduling components of a scientific workflow system to be able to operate, said components must be capable of subscribing to and interpreting real-time telemetry for both network and disk performance.

Exception Handling and Error Detection – An exception is an anomalous event, which occurs at runtime, e.g., disk, network, or service failure. Exceptions are handled (resolved) by: delineating an aspect of the workflow, such that any exceptions that are "raised" within said aspect are "caught" and propagated (or "bubbled"); inverting the flow of control; and, finally, restoring the software system to a consistent state [72]. Hence, exceptions are always known to the designer *a priori*. By contrast, an error is just an error, i.e., a mistake in the design and/or implementation of a software system, which can only be resolved by modifying said software system. Within the context of scientific workflows, prototypical errors include the use of services that are not fit for purpose, passing "incorrect" inputs to a given service, or failing to use all of the outputs of a given service. Hence, we identify three functional requirements for scientific workflow systems. First, scientific workflow systems should be fault-tolerant, and allow users to specify both the propagation and resolution strategies for checked exceptions, and the termination strategy for unchecked exceptions. Second, scientific workflow systems should homogenise exceptions that are raised during the performance of heterogeneous activities, e.g., implement a generic exception-handling mechanism, which can be specialised for specific exceptional behaviours. Finally, scientific workflow systems should allow users to annotate outputs that are intentionally unused.

Automatic Provenance Capture – Within the context of a scientific workflow system, provenance is a record of the derivation of a set of results. A key benefit of maintaining provenance information is that it provisions for the future reproducibility of said results, i.e., by providing a precise and unambiguous record of the sequence of computational steps that were followed in order generate each result, third-parties can reproduce and validate said results. Moreover, by maintaining provenance information, researchers can answer basic questions, including:

- Who created (and/or modified) this data item, and when?
- How was this data item generated?
- Do the elements of this set of data items share a common ancestor?

There are two distinct types of provenance [73]: prospective and retrospective. Prospective provenance encapsulates the specification of a formal process by which a set of results will be generated (in the future), i.e., the sequence of computational steps that will be evaluated. In contrast, retrospective provenance is a description of a formal process whose enactment resulted in the generation of a specific set of results (in the past), i.e., the sequence of computational steps that were evaluated, and the state of the environment both before and after the evaluation of each computational step.

We note that, the scientific workflow (as a whole) is a prospective provenance resource, which should be disseminated. Moreover, if said scientific workflow is developed according to an incremental methodology, then each successive revision, and the causal relationships between said revisions, are also examples of prospective provenance, which should also be disseminated. However, we note that scientific workflows are not self-evident, i.e., the dissemination of a scientific workflow does not constitute the dissemination of the rationale for the design and implementation of said scientific workflow, e.g., the "why" that explains the "what". Moreover, we note that, in a provenance-aware architecture, unless both a secure persistence mechanism and a non-repudiation strategy are provided, assertions of provenance information may be disputed. Hence, we identify three functional requirements for scientific workflow systems. First, scientific workflow systems should facilitate the automatic capture of both prospective and retrospective provenance information. Second, scientific workflow systems should provide a secure repository for provenance information. Third, scientific workflow systems should enforce a non-repudiation strategy for provenance information.

2.3.2 Provenance

Provenance, from the French *provenir*, "to come from," refers to an information resource, which asserts the origins of, and, hence, contextualises, a thing. In the context of computing, provenance refers to the record of the derivation of a unit of data, e.g., the agents that were associated with the derivation; the objects that were used as part of the derivation; and, the processes that were enacted as part of the derivation. It has been widely accepted [74] that the exposition of provenance information is essential for the establishment of trust in distributed software systems, such as the Web.

2.3.2.1 Perspectives on Provenance

The W3C PROV Model Primer [75] describes how provenance information can be viewed from at least three perspectives, where each perspective focuses on a different aspect of the information that is being described:

Agent-centred provenance describes conceptual entities that were "involved in generating or manipulating the information in question", e.g., the provenance information for a scholarly work may include a list of authors, contributors and editors.

Object-centred provenance describes the causal relationships between conceptual entities, e.g., the provenance information for a scholarly work, such as a conference paper, may include a list of references to external figures and datasets.

Process-centred provenance describes the "actions and steps taken to generate the information in question", i.e., process-centred provenance is object-centred provenance for agents.

In her musings, Goble [76] expands on the Zachman Framework [77], and argues that provenance is a representation of the seven W's:

$$\underbrace{\text{Who,}}_{\text{Agent}-\text{centred}}\ \underbrace{\text{What,}}_{\text{Object}-\text{centred}}\ \underbrace{\text{Where, When, (W)How,}}_{\text{Process}-\text{centred}}\underbrace{\text{Which and Why}}_{\text{Intent}-\text{centred}}$$

We note that the three perspectives are easily aligned with five of the seven W's. Moreover, we note that, by encapsulating the information content of each perspective as a distinct conceptual entity, it is possible to describe their inter-relationships (depicted in Figure 2.5).

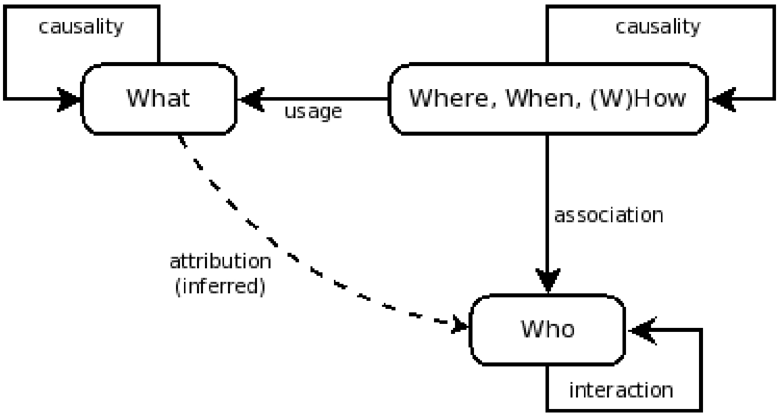


Figure 2.5: Depiction of relationships between conceptual entities that collectively encapsulate five of the seven W's of provenance.

Clearly, two agent-centred provenance descriptions are related if the conceptual entities that are being described interact (or have interacted) with one another. Similarly, object- and process-centred provenance descriptions are related if the conceptual entities that are being described exhibit (or will exhibit) a causal relationship. Moreover, a process-centred provenance description is related to an object-centred provenance description by way of some usage relationship, e.g., dependency, generation, modification, annihilation, etc. Similarly, a process-centred provenance description is related to an agent-centred provenance description by way of some association relationship, e.g., presence, instigation, action, etc. Finally, an object-centred provenance description is related to an agent-centred provenance description by way of some attribution relationship, which may be inferred.

But, what of the two remaining W's (which and why); questions of intent, whose answers are purely contextual, and hence, can be associated with any other type of provenance. To resolve this issue, we recall that there are in fact two types of provenance [73], prospective and retrospective, which correspond to the two tenses by which any information may be described, i.e., the future and past tenses respectively. We argue for the definition of a fourth perspective, "intent-centred" provenance, which describes the rationale and motivation for the specification of other provenance. Unlike the agent- and object-centred perspectives, intent-centred provenance information can be either prospective or retrospective. Hence, within this new framework, it is possible to align the two remaining W's.

2.3.2.2 Representation of Provenance

At the end of the day, provenance information is metadata (data about data). Hence, it is important that we understand the characteristics of this data, e.g., its representation; and, the nature of the relationship between the data that is being described, and the data that is providing the description [78]. For example, while any number of techniques may be used in order to collect provenance information, the information itself typically assumes one of two possible representations, either: annotation or inversion. In the former, assertions of provenance information take the form of reified conceptual entities, which relate some data to the information in question (referred to as "annotations"). In this case, we note that, along with the information in question, assertions exist *a priori*, and, hence, are readily available for use in provenance-oriented queries. In the latter, assertions of provenance information are, in principle, computable *a posteriori*, given a (mathematically) sufficient, process-oriented description of the information in question, which is said to be "invertible".

Furthermore, provenance information should have fixity, i.e., once it has been asserted, provenance information should be immutable. Hence, the conceptual entities that encapsulate provenance information are always "stateless", i.e., they have exactly one characteristic state, with zero outbound transitions. However, provenance information can be used in order to describe both the transitions between the characteristic states of "stateful" entities, and the immediate state of "stateless" entities. Therefore, we must be careful to distinguish between the two types of entities about which provenance information may be asserted.

For example, consider the provenance information that describes a temperature measurement, which was observed by a sensor in a specific location. At first glance, the conceptual entity that encapsulates the location appears to be "stateful", as its current temperature obviously changes over time. However, if one were to further develop this model, for instance, by defining a metadata term that denotes the current temperature, then the provenance information would be mutable, and, hence, would violate the requirement for fixity. By contrast, if a distinct conceptual entity is used in order to encapsulate each measurement, and said reifications are associated with the entity that describes the location, then all entities remain "stateless". Of course, the task of characterising the set of distinct states of any conceptual entity is domain-specific, and, hence, is beyond the scope of this discussion.

2.3.2.3 Applications of Provenance

By exposing provenance information about their resources, software systems are afforded a number of use cases. Goble [76] describes several applications of provenance information, including:

Provision of Context – Provenance affords additional context to the data that is being described. By exposing provenance information, users gain an enhanced understanding of the nature and characteristics of the data, which facilitates data exploration and discovery.

Attribution – Provenance establishes the authorship, ownership, and intellectual property rights for the data that is being described; enabling the citation of said data; and, if necessary, the determination of legal liability in the event that said data contains errors.

Measurement of Metrics – Provenance information is highly amenable to the measurement of metrics [79], such as: data quality and reliability; which can be used in order to automate process optimisation [80], or to estimate resource utilisation [81].

Establishment of Causality – Provenance enables the determination of causal relationships between objects and processes [82], e.g., objects are derived from one another, whereas the enactments of processes follow one another.

Justification and Replication of Results – Provenance provides a means for third parties to reproduce, replicate and validate results [83].

Within the context of scientific research, where, sadly, to this day, there are still too many cases of scientific misconduct, the capability to implement the final use case is of paramount importance. For example, in the biomedical- and life-sciences, scientific misconduct accounts for the majority of retracted publications [84]. Moreover, in domains that rely on data intensive processes, such as crystallography, it is possible for intermediate and derived data to be fraudulently modified or fabricated [85] .

The issue is succinctly summarised by Harrison, et. al., [86] who state that:

> “The falsified structures have many features in common: in each case, a *bona fide* set of intensity data, usually on a compound whose structure had been correctly determined and reported in the literature, was used to produce a number of papers, with the authors changing one or more atoms in the structure to produce what appeared to be a genuine structure determination of a new compound.”

Clearly, this is an issue of insufficient process- and object-centric provenance, i.e., if sufficient provenance information were provided, then third parties, such as the reviewers of scientific journals, would be able to validate each step of the reported process, and either: replicate the reported results; or, hopefully, discover the fraud.

2.3.2.4 Critique

We now proceed with our critique of exemplar vocabularies, taxonomies and ontologies for the exposition of provenance information.

Open Provenance Model (OPM) The OPM is a community-driven data model for the exposition of provenance information.

The development of the OPM is a direct result of the Provenance Challenge series [87], which was initiated in May 2006, at the first *International Provenance and Annotation Workshop*[17] (IPAW'06) workshop. The initial version of the OPM (v1.00) was released in December 2007 [88]. Following the completion of a community-driven evaluation process, the most recent version of the specification (v1.1) [89], was ratified and adopted in June 2009.

The design of the OPM is motivated by the following core requirements:

Semantic Alignment – To specify a common conceptual model and rule-set for the representation and inference of provenance assertions.

Data Integration – To facilitate the exchange of provenance information between heterogeneous software systems, i.e., to provide an interoperability layer.

[17] http://www.ipaw.info/ipaw06/

Due to its origins in the computational workflow community, the OPM is inherently process-centric. Furthermore, the OPM assumes that the retrospective provenance of any conceptual entity may be represented as a directed, acyclic graph (referred to as a "provenance graph"), whose nodes and edges correspond to actualised conceptual entities and causal dependencies respectively. Consequentially, it is not possible to use the OPM in order to assert prospective provenance.

In the OPM, provenance graphs are composed of three types of node:

Artefact[18] – A representation of an immutable state of an actualisation of a conceptual entity, i.e., an unchanging description of a pre-existing thing, which we may ask the provenance of.

Process – A representation of an action (or series of actions) that was (or were) performed in the past, whose execution resulted in the actualisation of new artefacts.

Agent – A representation of a conceptual entity who either: was the cause of; or, had an observable effect on, the past execution of a process.

In the OPM, provenance graphs are composed of five types of edge, which correspond to either: "timeless", binary- or timestamped, n-ary relations:

Generation – Denotes that an artefact was generated by a process;

Usage – Denotes that an artefact was used by a process;

Control – Denotes that an agent controlled the execution of a process (in an unspecified way);

Derivation – Denotes that an artefact was derived from another artefact; and

Communication – Denotes that the execution of a process was triggered by the execution of another process.

The Open Provenance Model Vocabulary (OPMV) is a codification of the OPM data model [90], which is divided into two parts: a core ontology (depicted in Figure 2.6), defined using OWL DL; and, a suite of supplementary modules, which provide additional, but less frequently used, metadata terms, along with specialisations of metadata terms from the core ontology.

We note that, according to the OPM, agents are disjoint to artefacts, i.e., agents are not artefacts. Hence, it is not possible to describe the retrospective provenance of "the controller" (an agent) as if it were an artefact. Specifically, it is not possible to describe

[18] In the OPM documentation, the American English variant "artifact" is used. However, for consistency, given the rest of this thesis, we maintain the British English spelling "artefact".

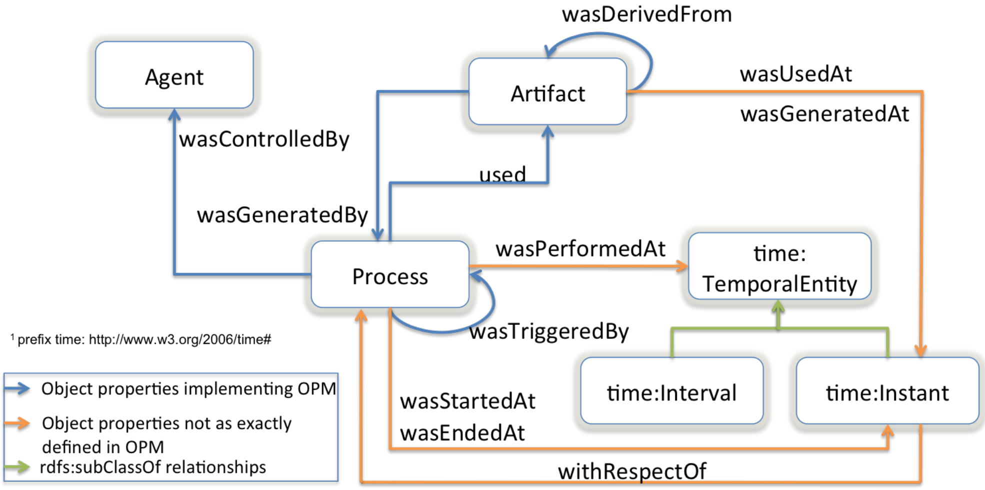


FIGURE 2.6: Depiction of entities and relationships in the core ontology for the Open Provenance Model Vocabulary (OPMV) (available at: http://open-biomed.sourceforge.net/opmv/img/opmv_main_classes_properties_3.png)

the retrospective provenance of situations where an agent was used as the input for a process, e.g., processes that elicit personal information. Similarly, it is not possible to describe the retrospective provenance of situations where the output of a process acted as the controller of subsequent processes.

PROV PROV is a family of specifications (depicted in Figure 2.7), developed by the W3C Provenance Working Group[19], which collectively define a data model and accompanying methodology for the exposition and exchange of provenance information [91].

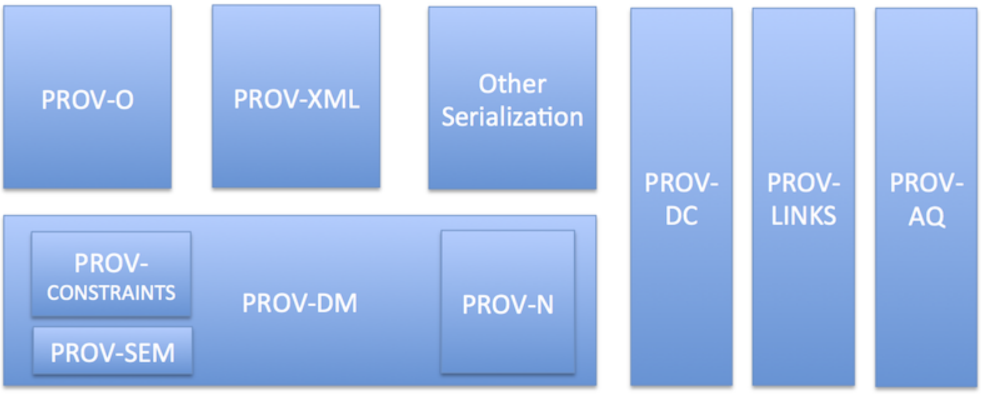


FIGURE 2.7: Depiction of the organisation of PROV specifications (available at: http://www.w3.org/TR/2013/WD-prov-overview-20130312/prov-family.png)

Citing the OPM as one of its main references, the initial working drafts of the PROV specifications were published in July 2012.

[19] http://www.w3.org/2011/prov/wiki/Main_Page

Due to the less restrictive semantics of its data model, PROV facilitates the exposition of agent-, object-, or process-centred provenance information. In contrast to the OPM, which defines a conceptual entity in order to represent each "Artefact", PROV defines an conceptual entity in order to represent each "Entity", which may be any thing, real or imaginary, stateful or stateless.

We note that, according to the PROV data model, agents may be declared as specialisations of both entities and activities. Therefore, unlike OPM, it is possible to use PROV in order to assert the provenance of the agents themselves.

Finally, we note that, unlike OPM, it is possible to use PROV in order to assert prospective, process-centred provenance, i.e., to assert the intention of an agent that is associated with an activity. Specifically, the PROV data model specifies the "Plan" conceptual entity, which represents "a set of actions or steps intended by one or more agents to achieve some goals" [92]. In theory, the inclusion of a plan conceptual entity should facilitate the exposition of prospective provenance, however, in practice, the PROV data model does not define the characteristics of plans. Therefore, other than deciding whether or not a plan has been specified, no meaningful analysis can be conducted of the observed relationships between the intentions and actions of each agent. An extension to PROV has been proposed [93], which augments the definition of a plan, in order to remedy this issue.

Ontology of Scientific Experiments (EXPO) The goal of EXPO is to provide a common ontology for the formal description of the design, methodology and outcomes of a scientific experiment, which is defined as "a research method which permits the investigation of cause-effect relations between known and unknown (target) variables of the domain" [94].

EXPO is defined using OWL DL, as an extension of the Suggested Upper Merged Ontology (SUMO) [95], an upper ontology, which is intended to be used as a foundation ontology, providing an interoperability layer for domain-specific ontologies.

The core of EXPO is the "ScientificExperiment" class; a specialisation of the SUMO "Process" class, which is defined as a retrospective description of a sequence of events that have happened (in the past). Hence, EXPO is suitable for the exposition of retrospective provenance information. Moreover, EXPO includes a "PlanOfExperimentalActions" class; a specialisation of the SUMO "Plan" class, which is defined as a specification for a sequence of processes. Thus, EXPO is also suitable for the exposition of prospective provenance information. Finally, we note that EXPO uses the SUMO "has-plan" predicate in order to relate a retrospective description of a process to the prospective description of its plan.

Ontology of Data Mining Investigations (OntoDM) The goal of OntoDM is to provide a common ontology for the formal description of information resources from the domains of data mining and knowledge discovery, and to facilitate the assertion of relations between "entities that are realized [sic] in a process or processes" [96].

OntoDM is defined using OWL DL, as an extension of the Ontology for Biomedical Investigations (OBI) [97], which is itself aligned with the Basic Formal Ontology (BFO) [98]. Both OBI and the BFO are upper ontologies, which are intended to provide an interoperability layer for domain-specific ontologies.

The decision to align with OBI, and hence, with BFO, rather than with SUMO, is an acknowledgement, by the designers of OntoDM, of the generality of formal processes, and the domain-specificity of data mining investigations. BFO is grounded by the concept of "realisation", and hence, defines generic conceptual entities for the representation of "plans", "processes", and "planned processes" (processes that are associated with one or more plans). This is complemented by OBI, which defines conceptual entities for specific domains of discourse, including: physics, chemistry, biology, etc; along with specialised predicates for relating "realisable" and "realised" entities. Given this basis, we note that OntoDM is suitable for the exposition of both prospective and retrospective provenance information.

2.3.3 Summary

Given our analysis, we draw the following conclusions:

1. The use of computational workflows is facilitated by advanced data representation, management and integration capabilities.
2. The transition towards data-intensive science necessitates the specification of methodologies for the capture, curation and dissemination of both prospective and retrospective provenance information.
3. We argue for the definition of a fourth perspective for the specification of provenance information: intent-centred provenance, which captures the rationale and motivation for the specification of other provenance information.
4. The concept of "actualisation" (or "realisation") can be used in order to relate the retrospective provenance of an action (that was performed) with the prospective provenance of the intention to perform said action.

Chapter 3

Chemistry on the Semantic Web

At the beginning of this dissertation, we outlined the requirement for advanced data integration capabilities in scientific software systems. This requirement is to be able to unambiguously interpret conceptual entities, which are common to multiple domains, such that data may be shared and re-used. In this chapter, we begin to address this requirement by introducing three new datasets. The aim of these datasets is to identify and relate conceptual entities that are relevant to many sub-domains of chemistry, and would therefore, benefit from standardisation. We also describe and implement a real-world application, which makes immediate use of the new datasets: a health and safety assessment form generator.

The goal of this work is not only to demonstrate the pragmatic construction and re-use of the new datasets, but also to acquire an understanding of the goals and needs of the laboratory-based researcher. Without this understanding, it is impossible to provide a compelling value-proposition for the deployment of knowledge representation techniques and technologies in the laboratory. In order to provide a viable solution, which not only meets the functional requirement of facilitating data integration, but also meets the non-functional requirements of extensibility and repurposability, it is necessary to understand and answer several questions regarding the construction of each dataset. To do this, we define a methodology for the construction of each dataset that is informed by the medium by which its conceptual entities are presented. With these methodologies, we then proceed to construct the datasets.

The contributions of this chapter are as follows:

1. A controlled vocabulary for quantities, units and symbols that are used in physical chemistry;
2. A controlled vocabulary for the classification and labelling of potentially hazardous chemical substances;

3. A Linked Data interface for the RSC ChemSpider online chemical database; and
4. An automated, legally compliant health and safety assessment form generator.

The remainder of this chapter is organised as follows. First, a controlled vocabulary for quantities, units and symbols in physical chemistry is presented. Second, a controlled vocabulary for the classification and labelling of hazardous chemical substances is presented. This is followed by the presentation of a Linked Data interface for the RSC ChemSpider online chemical database. After that, a health and safety assessment form generator, which makes pragmatic use of the new datasets, is presented. Finally, conclusions are drawn.

3.1 IUPAC Nomenclature

A nomenclature is a system for the assignment of names to conceptual entities. By agreeing on the use of a specific nomenclature, individuals within a network may assign names to conceptual entities, and make reference to those names when communicating with other individuals in said network. The utility of this approach is that, in essence, each nomenclature captures a mathematical function, whose domain is a set of ambiguously identified conceptual entities, and whose co-domain is a set of unambiguously identified conceptual entities, i.e., each nomenclature is a distinct resolution mechanism, where the elements of the co-domain are inverse-functional identifiers.

A chemical nomenclature is a system for the assignment of names to chemical compounds. The primary role of a chemical nomenclature is to specify a non-injective and surjective mapping, which may be used to unambiguously identify the chemical compound, i.e., to define an algorithm for naming each chemical compound according to a system of well-defined rules. Within the context of a chemical nomenclature, it is assumed that two chemical compounds are the same if and only if they map to the same name.

The International Union of Pure and Applied Chemistry (IUPAC)[1] develops and maintains the most widely used chemical nomenclature – IUPAC nomenclature – as a series of publications, which are commonly referred to as the “coloured books” (the cover of each book is printed using a distinct colour). Each book is targeted at one aspect or sub-domain of chemistry research:

Blue Defines naming conventions for organic chemical compounds.

Red Defines naming conventions for inorganic chemical compounds.

[1] http://www.iupac.org/

Green Provides recommendations for the use of terminology, units of measurement and symbols in physical chemistry.

Gold Defines [general] terminology for use in chemistry.

White Defines terminology for use in biochemistry.

Orange Defines terminology for use in analytical chemistry .

Purple Defines terminology for use in macromolecular chemistry.

Silver Defines terminology for use in clinical chemistry.

The first IUPAC manual of Symbols and Terminology for Physiochemical Quantities and Units (the "Green Book") was published in 1969, with the goal of "securing clarity and precision, and wider agreement in the use of symbols by chemists in different countries" [99]. In 2007, following an extensive review process, the third (and most recent) edition of the Green Book was published.

In total, the third edition of the Green Book aggregates 250 A4-sized pages of content, of which:

- 14 pages are labelled with Roman numerals;
- 233 pages are labelled with decimals; and
- 3 pages are unlabelled.

The primary goal of this work is to construct a controlled vocabulary from the subject index of the third edition of the Green Book. To achieve this goal, we first construct a model of the subject index. We continue by populating our model using an automated software application. Finally, we transform the populated model into a controlled vocabulary using a deterministic algorithm.

The secondary goal of this work is to assess the overall quality of the subject index using quantitative methods. We work towards this goal by answering the following questions:

Coverage Does the subject index reference every page of the main body of the text?

Distribution What is the frequency distribution (with respect to pages) of references in the subject index?

When compiling the subject index for a physically-delineated text (where regions of text are separated by physical boundaries, such as the edge of a page), good coverage is essential. In real terms, this means two things: For every page of the text, there exists at least one reference (page coverage); and, for every page of the text, for every concept that is discussed by that page, there exists at least one reference (term coverage). As we are not domain specialists, we are unable to construct the set of conceptual entities that are discussed by each page of the text, therefore, we are unable to make assertions about term coverage. However, we are still able to make assertions about page coverage.

The second quality of the subject index that we assess is the frequency distribution of references (with respect to pages). For a thoughtfully-constructed subject index, the frequency distribution should correlate roughly with the high-level structure of the text, i.e., the chapters and sub-chapters. Conversely, if we find no correspondence between the frequency distribution and the high-level structure of the text, then we may infer that the subject index is badly constructed.

The remainder of this section is organised as follows. First, we present our methodology for modelling the subject index of the Green Book. Second, we present our methodology for extracting and enriching the information content of the subject index of the Green Book. This is followed by a presentation of our methodology for the construction of a controlled vocabulary. Finally, conclusions are drawn regarding the new dataset.

3.1.1 Methodology for Modelling of Subject Indices

In this section, we present our methodology for the modelling of the subject index of the Green Book, and the derived entity-relationship model.

ab initio, 20
abbreviations, 157–164
abcoulomb, 139
absolute activity, 46, **57**
absolute electrode potential, 71
absorbance, **36**, 37, 39, 40
  decadic, 5, **36**, 37
  napierian, **36**, 37

```
\item ab initio, 20
\item abbreviations, 157--164
\item abcoulomb, \uu{139}
\item absolute activity, 46, \bb{57}
\item absolute electrode potential, 71
\item absorbance, \bb{36}, 37, 39, 40
     \subitem decadic, 5, \bb{36}, 37
     \subitem napierian, \bb{36}, 37
```

Figure 3.1: Excerpt of the subject index of the third edition of the IUPAC Green Book (left) and corresponding LaTeX source (right).

An excerpt of the subject index of the Green Book and its corresponding LaTeX source are given in Figure 3.1. We make the following observations:

- The subject index is presented as a tree structure;
- The maximum depth of the tree is 1;
- The nodes at each level of the tree are listed in alphabetical order (Greek letters are ordered according to their British English spelling and accented characters are ignored);

- The label for a child node is shortened by removing the label of the parent node, e.g., "decadic absorbance" is listed as "decadic";
- Each term may assert zero or more references;
- The references for each term are listed in numerical order;
- When more than one reference is given, **bold** print is used to indicate the general (or defining) reference and underlining is used to indicate a numerical entry (where the term identifies the corresponding physical quantity);
- References to an inclusive range of pages (specified by a first and last page) are separated by a hyphen symbol "–"; and
- References may span multiple lines of the LaTeX source (the additional line-break characters are not included in the typeset subject index).

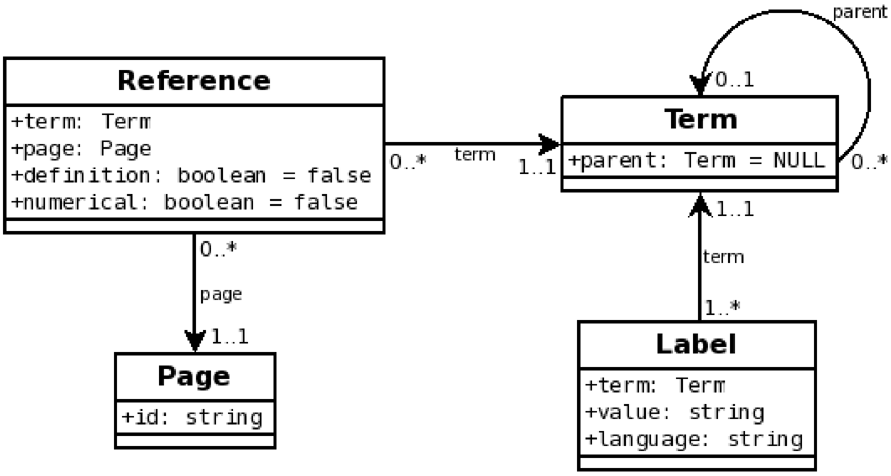


FIGURE 3.2: UML class diagram for the proposed entity-relationship model of the subject index of the third edition of the IUPAC Green Book.

The UML class diagram for our entity-relationship model of the subject index is presented in Figure 3.2. The model describes four entity types:

Term A keyword that captures the essence of a topic in the text. The tree structure of the subject index is captured by the "parent" relationship, which relates each term to zero or one other term, i.e., a term is not allowed to be its own parent.

Label A string of characters that may be used as the identifier for a term. This entity is included in order to facilitate internationalisation of the dataset, i.e., each term may have one or more labels, where each label is associated with a distinct language.

Page A page of the text, which is associated with a unique identifier (either a decimal or a Roman numeral).

Reference A reification, which relates exactly one term to exactly one page. Range references from page m to page n, where $m \leq n$, are modelled as $(n - m) + 1$ distinct references.

3.1.2 Methodology for Extraction and Enrichment of Subject Indices

In this section, we present our methodology for the extraction of the subject index of the Green Book. We also describe the population of our entity-relationship model, and the enrichment of the resulting dataset.

To produce the third edition of the Green Book, the editors selected the LaTeX typesetting system[2]. From our perspective, this was a fortuitous choice, as the resulting document is highly amenable to established data mining techniques, such as text recognition.

To typeset the subject index, the editors selected the `theindex` environment [100, pp. 54]. Each term is declared with an `\item` command. A sub-term is declared with `\subitem`, and a sub-sub-term is declared with `\subsubitem`. An extra vertical space is introduced by the `\indexspace` command, which is used to separate the first entry for each letter.

```
theindex    : ‘\begin{theindex}’ ( item | ‘\indexspace’ )* ‘\end{theindex}’ ;
item        : ‘\item’ term ( subitem )* ;
subitem     : ‘\subitem’ term ( subsubitem )* ;
subsubitem  : ‘\subsubitem’ term ;
term        : TERM ( ‘,’ rangeRef )* ;
rangeRef    : reference ( ‘--’ reference )? ;
reference   : PAGE | ‘\bb{’ reference ‘}’ | ‘\uu{’ reference ‘}’ ;
```

FIGURE 3.3: Non-terminal production rules of a grammar (in ANTLR v3 syntax) whose corresponding parser recognises indices that were generated by the `theindex` environment for LaTeX.

In Figure 3.3, we give the non-terminal production rules for a grammar, in ANTLR v3[3] syntax, whose corresponding parser recognises indices that were generated by the `theindex` environment. The grammar defines three terminal production rules: `TERM` and `PAGE`, which denote the labels for an index term and a page of the text respectively; and `WS`, which denotes white-space characters (ignored by the parser).

As it describes a sequence of unquoted, multi-token strings of non-standard characters, significant difficulties were encountered when attempting to “convince” ANTLR v3 to accept the validity of the `TERM` production rule. Accordingly, we decided to manually

[2] http://www.latex-project.org/

[3] http://antlr.org/

implement the grammar as a bespoke software application, written using the Java programming language[4].

Algorithm 1 Binary function for normalisation of labels for terms that are extracted from the subject index of the third edition of the IUPAC Green Book.

```
function NORMALISE($l_n$, $l_{n+1}$)                    ▷ labels for nodes at depth $n$ and $n+1$
    if $endsWithAny\left(l_n,\ \{`\ for',\ `\ of'\}\right)$ then return $concat\left(\{l_n,\ `\ ',\ l_{n+1}\}\right)$
    else
        if $endsWithAny\left(l_{n+1},\ \{`-'\}\right)$ then return $concat\left(\{l_{n+1},\ l_n\}\right)$
        else return $concat\left(\{l_{n+1},\ `\ ',\ l_n\}\right)$
        end if
    end if
end function
```

When executed, the software application recognises each line of the input, and, in accordance with the grammar, persists the appropriate records in a temporary, in-memory database. It then proceeds to normalise the set of extracted labels (Algorithm 1), and to relate terms based on their pairwise cosine similarities. Finally, the contents of the database are exported as a serialisation of an RDF graph.

To calculate the cosine similarity $f(A, B)$ between two terms A and B, we construct an n-bit vector for each term, where n is the total number of pages, and the truth of the k^{th} bit corresponds to the presence of a reference to the k^{th} page (Equation 3.1).

$$f(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\,\|B\|} \tag{3.1}$$

The range of the cosine similarity function is between zero and one (inclusive). We interpret the result as follows:

$f(A, B) = 0 \longrightarrow A$ and B are <u>never</u> discussed by the same pages.

$0 < f(A, B) < 1 \longrightarrow A$ and B are sometimes discussed by the same pages.

$f(A, B) = 1 \longrightarrow A$ and B are <u>always</u> discussed by the same pages.

Cosine similarities are calculated because their presence (or absence) provides the basis for the assertion of non-hierarchical relationships between terms in the subject index. Other than the enrichment of the dataset, an advantage of this approach is that, if a suitable coefficient is selected to represent "similar" pairs, e.g., unity, then the resulting assertions can be used as a navigational aid. Hence this approach is particularly well-suited to the IUPAC Green Book, as "similar" pairs (of terms in the subject index) correspond to "related" concepts (of the same subdomain of discourse).

[4] http://java.sun.com/

3.1.3 Methodology for RDF Representation of Subject Indices

In this section, we present our methodology for the representation of the subject index of the Green Book as an RDF graph.

Prefix	Namespace URI	Description
`rdf`	http://www.w3.org/1999/02/22-rdf-syntax-ns#	RDF vocabulary terms
`skos`	http://www.w3.org/2004/02/skos/core#	SKOS vocabulary terms

Table 3.1: Namespaces and prefixes used in Section 3.1.3.

We begin by creating a new URI to uniquely identify an RDF resource that describes the Green Book, and may be used in order to assert bibliographic metadata using Dublin Core:

`http://id.iupac.org/publications/iupac-books/161`

Next, we create a new URI to uniquely identify an RDF resource that describes the subject index of the Green Book:

`http://id.iupac.org/publications/iupac-books/161/subjects`

We model the subject index as an instance of `skos:ConceptScheme`, and each term as an instance of `skos:Concept` (instances are associated with their types by asserting the `rdf:type` predicate). Lexical labels are associated with each term using the `skos:prefLabel` predicate.

The URI for the RDF resource that describes each term is constructed by appending the URL-encoded representation of the British English translation of its label to the URI of the subject index[5]:

`http://id.iupac.org/publications/iupac-books/161/subjects/<Label>`

Finally, we make the following assertions, which collectively encode the tree-structure of the subject index within the resulting RDF graph:

- Each RDF resource that describes a term is related to the RDF resource that describes the subject index by asserting the `skos:inScheme` predicate.
- RDF resources that describe sub-terms and sub-sub-terms are associated with their ancestors by asserting the `skos:broader` predicate (and its inverse, the `skos:narrower` predicate).

[5] It should be noted that, in this system, all URIs should be treated opaquely, i.e., one should not infer anything from their structure. Instead, information, such as the label, should be obtained by dereferencing the URI for the description of the term, and by subsequently analysing the response.

- RDF resources that describe root terms (those with no ancestors) are associated with the RDF resource that describes the subject index itself by asserting the `skos:topConceptOf` predicate (and its inverse, the `skos:hasTopConcept` predicate).
- RDF resources that describe terms which are similar to each other are associated by asserting the `skos:related` predicate. For simplicity, we assume that two terms A and B are similar if and only if their cosine similarity is unity $f(A, B) = 1$.

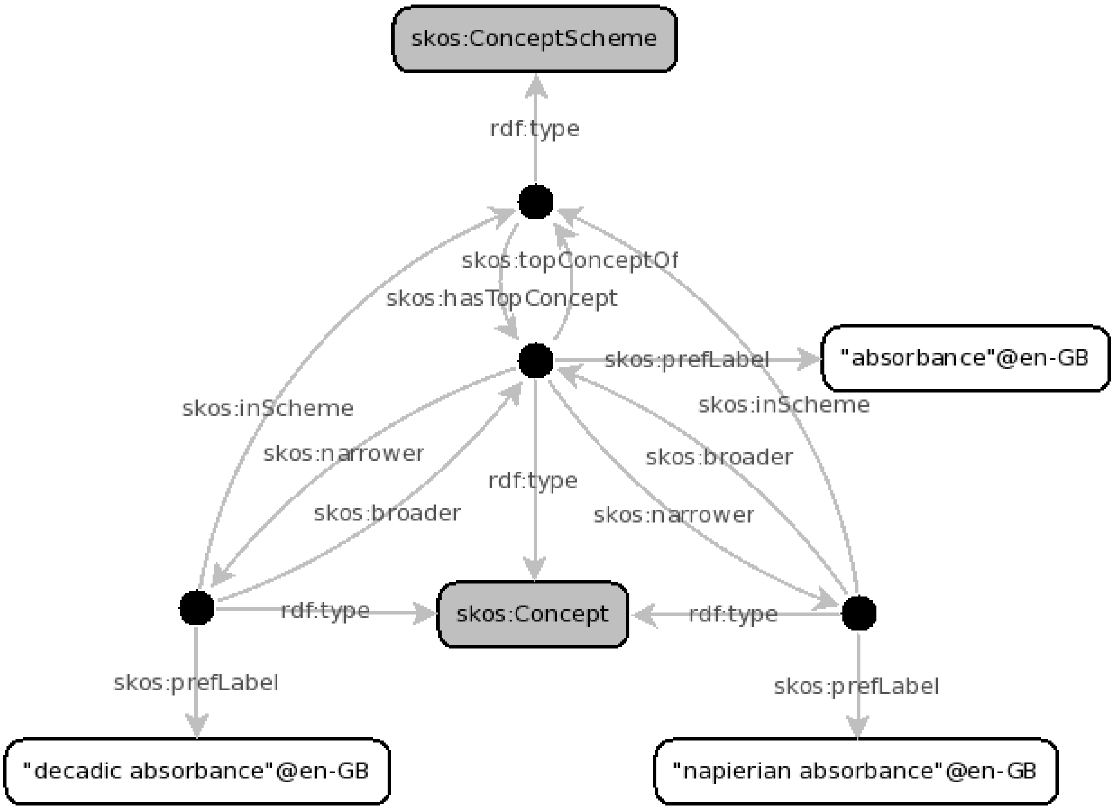


FIGURE 3.4: Depiction of RDF graph that describes three terms from the subject index of the IUPAC Green Book.

In Figure 3.4, we give a depiction of a region of the resulting RDF graph, which describes three terms from the subject index of the Green Book: “absorbance”, “decadic absorbance” and “napierian absorbance”.

3.1.4 Summary of Dataset

In this section, we present a summary of the dataset that was generated from the subject index of the third edition of the IUPAC Green Book.

In total, we extracted 4101 references to 2490 distinct terms, which were spread over 155/250 pages of the text. Despite the fact that only 62% of pages are referenced, we still found that the subject index has excellent page coverage. All unreferenced pages can be accounted for, as they are either: part of the preface, bibliography, index of symbols, or subject index; or, are one of the many separator pages between chapters and major sections, which are "intentionally left blank".

During the data enrichment phase, 66079 non-zero cosine similarity pairs were discovered, of which 14154 pairs had a similarity coefficient of unity. Given the total number of terms in the subject index, the maximum number of distinct pairs is $2490^2/2 = 3100050$. Therefore, we can assert that, within the context of our methodology, around 2.132% of terms are related by a non-zero cosine similarity, and 0.457% are related by a cosine similarity of unity.

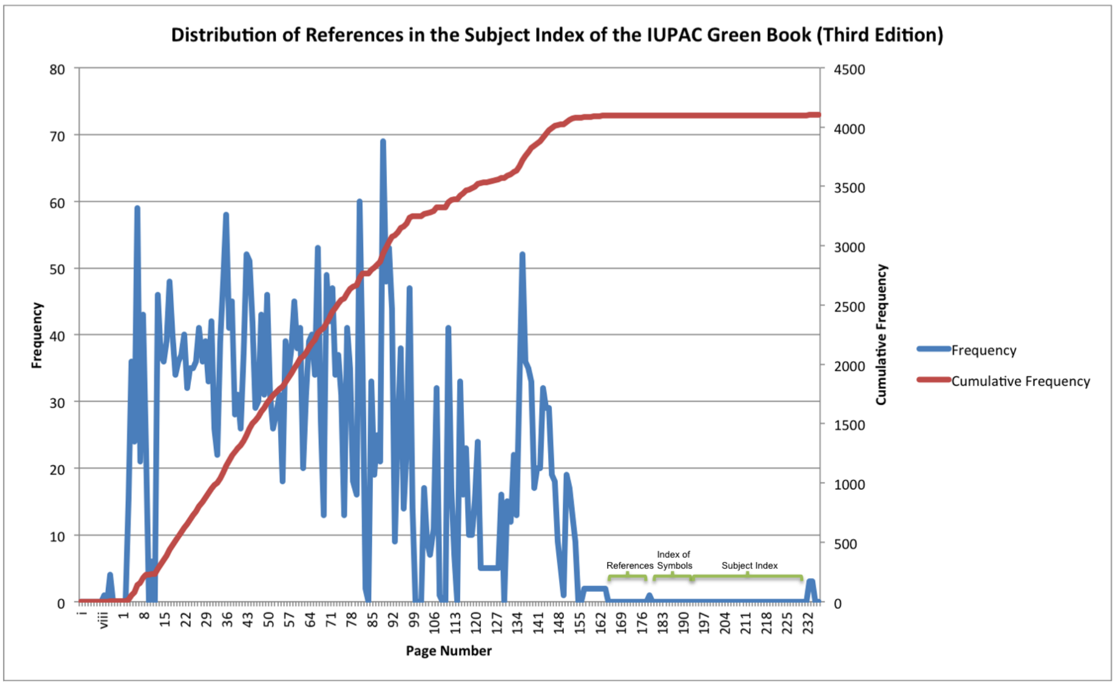
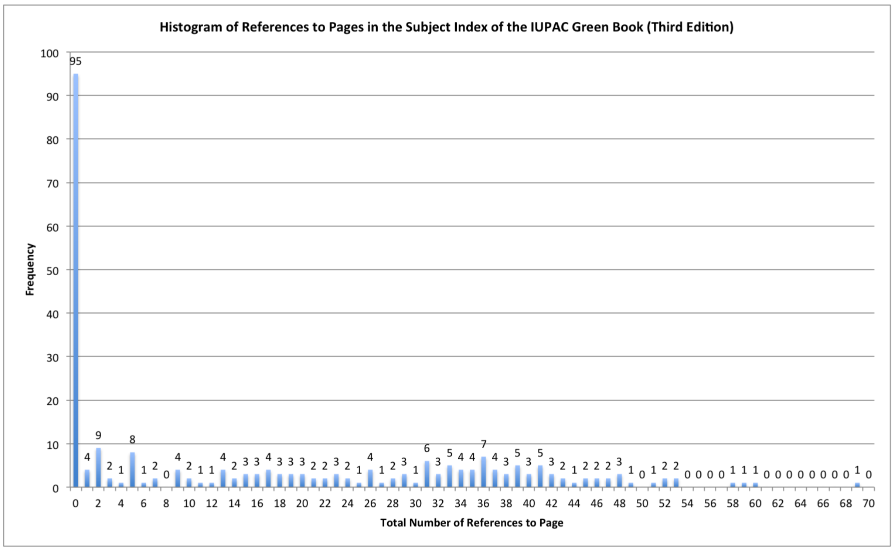


Figure 3.5: Distribution of references in the subject index of the third edition of IUPAC Green Book.

In Figure 3.5, we give the frequency distribution (with respect to pages) for references that were extracted from the third edition of the subject index of the Green Book. We make two key observations: First, the curve that depicts the frequency distribution has multiple, distinct peaks, i.e., it has a high level of peakedness. Each peak is the local maxima of a contiguous range of pages. Visual inspection of the text indicates that each of these ranges corresponds to a high-level section, e.g., a chapter or sub-chapter, where the local maxima is the most prominent page in the range. Second, the curve that depicts the cumulative frequency plateaus between pp165–250. Within this range, there are two very minor peaks (of less than 10 references), which correspond to the table of symbols for letters of the Greek Alphabet (pp179) and the table of numerical energy conversion factors (pp234). Given that the remainder of the plateau is accounted for by the bibliography, index of symbols and subject index, we may infer that there is a high degree of correlation between the frequency distribution of references (with respect to pages) and the high-level structure of the text.



	Mean	Median	Mode	Std. Deviation	Variance	Range
Incl. Zeros	16.316	7	0	17.886	319.899	69
Excl. Zeros	26.477	28	2	15.783	249.108	68

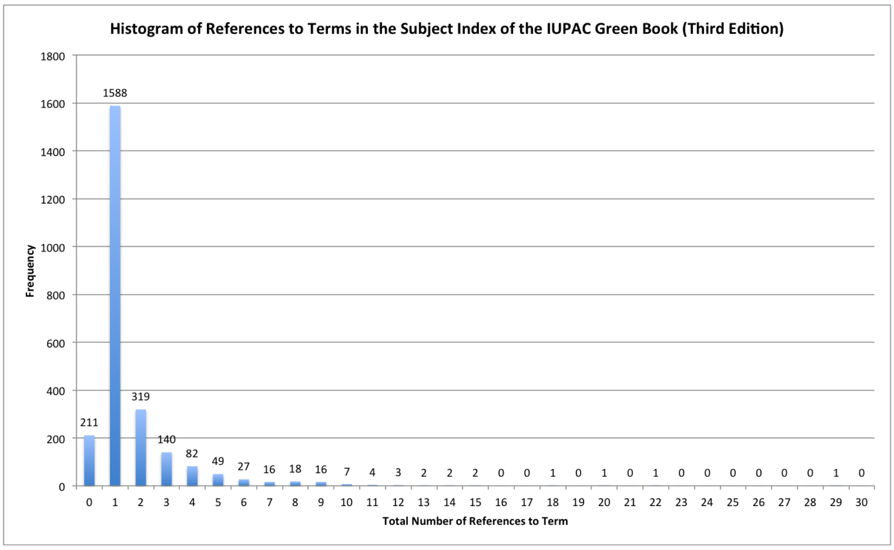
FIGURE 3.6: Histogram of total number of references to pages in the subject index of the third edition of the IUPAC Green Book.

In Figure 3.6, we give the histogram for the total number of references to each page in the third edition of the Green Book, along with a table of descriptive statistics. When unreferenced pages are included in the calculations, we find that the arithmetic mean is approximately 16.316 (references per page), with a standard deviation of 17.886 (references per page). In contrast, when unreferenced pages are excluded, we find that the arithmetic mean is approximately 26.477 (references per page), with a standard deviation of 15.783. The high degree of variability in the data is explained by the influence of the 95 unreferenced pages on the value of the arithmetic mean.

In Figure 3.7, we give the histogram for the total number of references to each term in the third edition of the Green Book, along with a table of descriptive statistics. The data shows little variation regardless of the inclusion of unreferenced terms in the calculations. This is explained by the fact that the around 64% of the terms are only referenced once by the subject index.

In Figure 3.8, we present the weighted list (or "tag cloud") of the most frequently referenced terms in the subject index of the third edition of the Green Book (rendered by Wordle[6]). The weight of each element in the list is proportional to the number of

[6] http://www.wordle.net/



	Mean	**Median**	**Mode**	**Std. Deviation**	**Variance**	**Range**
Incl. Zeros	1.648	1	1	1.896	3.594	29
Excl. Zeros	1.801	1	1	1.911	3.652	28

FIGURE 3.7: Histogram of total number of references to terms in the subject index of the third edition of the IUPAC Green Book.

references for the corresponding term in the subject index. The advantage of presenting the subject index as a weighted list is that it is trivial for non-domain-specialists to identify the most prominent elements, and to determine their relative prominence (with respect to other elements).

Term	**Frequency**	**Term (cont.)**	**Frequency**
mass	29	solution	12
length	22	electric field strength	11
energy	20	elementary charge	11
ISO	18	frequency	11
IUPAC	15	speed of light	11
atomic unit	15	angular momentum	10
IUPAP	14	base unit	10
time	14	concentration	10
amount of substance	13	second	10
temperature	13	spectroscopy	10
force	12	unified atomic mass unit	10
physical quantity	12	wavenumber	10

TABLE 3.2: Terms from the subject index of the third edition of the IUPAC Green Book with 10 or more references (terms with the same frequency are given in alphabetical order).

In Table 3.2, we give a list the most frequently referenced terms in the subject index, which correspond to the most prominent elements of the weighted list in Figure 3.8.

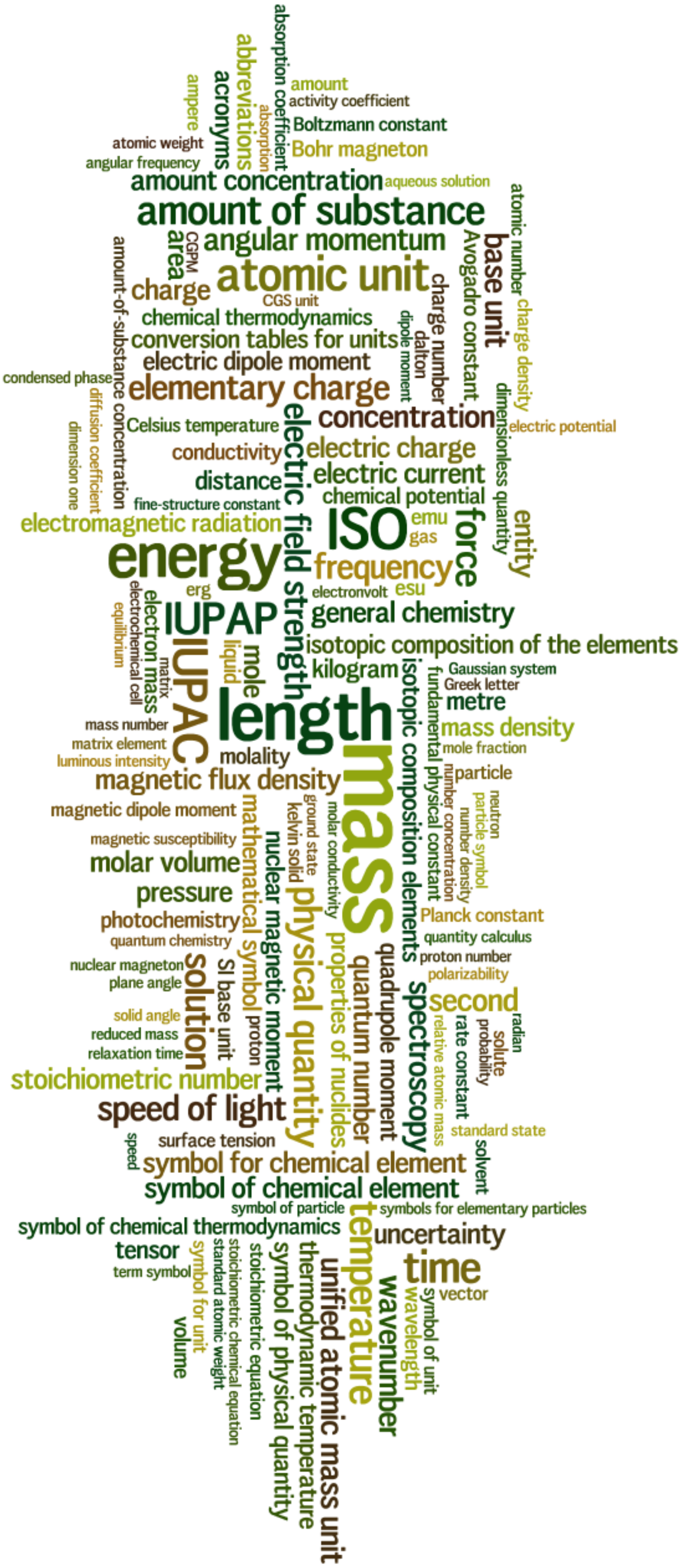


Figure 3.8: Depiction of weighted list (or “tag cloud”) of most frequently referenced terms in the subject index of the third edition of the IUPAC Green Book.

Finally, in total, the dataset describes an RDF graph of 40780 triples.

3.2 Globally Harmonized System of Classification and Labelling of Chemicals (GHS)

The Globally Harmonized System of Classification and Labelling of Chemicals (GHS) is an internationally agreed-upon system for the classification and labelling of chemical substances and mixtures, which was created by the UN in 2005. As its name suggests, the GHS is intended to replace (or "harmonise") the various systems for classification and labelling that are currently in use around the world, with the goal of providing a consistent set of criteria for assessment, which may be re-used on a global scale. The manuscript for the GHS, which is published by the UN, is commonly referred to as the "Purple Book" [101][7].

Before the creation of the GHS, there were many competing systems, which were in use in different countries. Whilst these systems all satisfied the same set of functional and non-functional requirements (to facilitate the classification and labelling of chemical substances and mixtures), they did so in different ways, creating an environment where the classification and labelling entities were ambiguously identified. Given the dramatic growth of the chemicals export sector in recent years, and the potential for negative impact on human health and the natural environment in countries where proper controls are not implemented, it was decided (by the UN) that a global system was necessary.

Following the international agreement of the GHS, the European Union (EU) proposed the Regulation on Classification, Labelling and Packaging of Substances and Mixtures (CLP), which is commonly referred to as the "CLP Regulation" [102]. The CLP Regulation was published in the official journal of the EU on 31 December 2008.

The CLP Regulation entered into legal effect in all EU member states on 20 January 2009, and was immediately subjected to an extended transitional period. In accordance with EU procedure, the provisions of the CLP Regulation will be gradually phased into law over a period of years, until 1 June 2015, when it will be fully in force for both substances and mixtures.

The CLP Regulation aggregates 1355 A4-sized pages of content. The main body of the document introduces the GHS classification and labelling entities, and outlines the rules for the application of the legislation. The remainder of the document is subdivided into the following annexes:

[7] Presumably, the manuscript for the GHS was named for a specific colour in accordance with the publishing practices of IUPAC, however, it is not known why the UN committee selected the colour purple, which is already used for the cover of the IUPAC Compendium of Macromolecular Chemistry. In the opinion of the author, a better choice would have been luminous pink, which has the key advantages of (a) not being in use for any other IUPAC coloured book; and (b) being eye-catching and visually distinctive, making the manuscript easier to locate in a wet laboratory environment.

Annex I – defines the criteria for classification and labelling of chemical substances and mixtures; and defines the hazard classes and their differentiations;

Annex II – defines "special rules" for labelling and packaging of certain chemical substances and mixtures; and defines EU-specific hazard statements;

Annex III – provides a complete list of hazard statements, along with translations for all 23 official and working languages of the EU;

Annex IV – provides a complete list of precautionary statements, along with translations for all 23 official and working languages of the EU;

Annex V – provides a complete list of hazard pictograms;

Annex VI – lists hazardous substances and mixtures for which harmonised classification and labelling have been established;

Annex VII – provides a translation table from classification under Directive 67/548/EEC to classification under the CLP Regulation.

Currently, the full text of the CLP Regulation is available online as a PDF document[8], which, unfortunately, is not amenable to machine processing, and therefore, is non-trivial to incorporate into existing software applications. Thus, there is a need for the information content of the CLP Regulation to be made available in a machine-processable format, such as RDF, where the classification and labelling entities are unambiguously identified. To achieve this goal, we first construct a model of the concepts that are defined in the main body of the CLP Regulation. We continue by populating the model using the "instances" that are defined in the annexes of the CLP Regulation.

The remainder of this section is organised as follows. First, in order to gain a broad understanding the problem, we describe the purpose of a classification and labelling system. Second, we describe our methodology for modelling the information content of the CLP Regulation, and present our derived entity-relationship model. This is followed by our methodology for representing the information content of the CLP Regulation as an RDF graph. Next, we describe a web application that presents the dataset as a human- and machine-readable knowledge base. Finally, conclusions are drawn regarding the new dataset.

3.2.1 To Distinguish Between Hazardousness and Riskiness (The Purpose of a Classification And Labelling System)

A hazard class (or simply, a hazard) is a conceptual entity that denotes a quality of a chemical substance. Each hazard class represents the possibility of the realisation of an

[8] http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2008:353:0001:1355:EN:PDF

observable phenomena, e.g., “flammable”. A hazard category is a combination of a hazard class and an adverb, which specifies relative likelihood of occurrence (or “severity”), e.g., “[no adverb] flammable”, “highly flammable”, and “extremely flammable”.

Hazard classes are unordered, however, hazard categories of the same class are ordered. The order of two hazard categories (of the same class) is determined by associating each of their adverbs with a representational measure, such as a real number, which denotes the relative likelihood of the realisation of the phenomenon, and is denoted by the hazard class (referred to as “hazardousness”), e.g., “[no adverb]” = 1, “highly” = 2, and “extremely” = 3.

The degree of risk (referred to as “riskiness”) is a measure of the likelihood of the realisation of a causal relationship between the realisation of the phenomena that is denoted by a hazard class (the cause) and the realisation of an undesirable event (the effect), within the context of a specific working environment. Thus, the riskiness of a hazard is dictated by our ability to mitigate its effects, within the context of the working environment, and not by the hazardousness of the effects themselves.

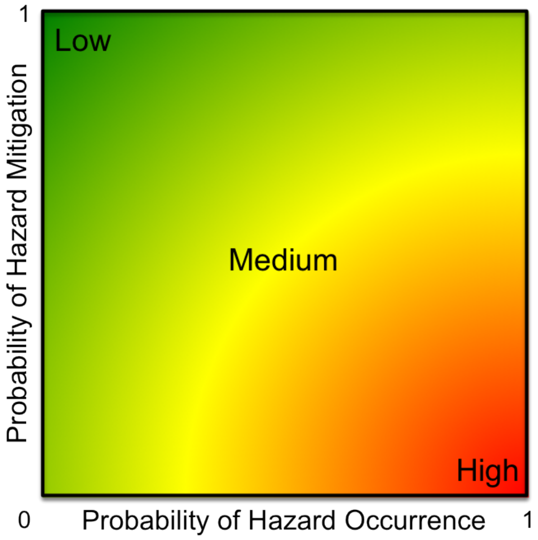


Figure 3.9: Depiction of the riskiness of a hazard whose risk assessment function is defined by a linear gradient. The “traffic light” colouring system is used to indicate the magnitude of the codomain at each point.

Typically, the riskiness of a hazard is determined as the result of a risk assessment; a binary mathematical function (referred to as the “risk assessment function”), which associates a representational measure, such as a real number, with each pair of likelihoods (for occurrence and mitigation). In Figure 3.9, we give a depiction of an exemplar risk assessment function that is defined by a linear gradient. The exemplar captures the most common pattern for risk assessment: the riskiness of a specific hazard class varies linearly with our ability to mitigate its effects. For example, the risk of exposure when handling chemical substances that give off extremely toxic vapours is “high” when the experiment is conducted inside a sealed room, and “low” when the experiment is conducted inside a fume cupboard. Thus, each risk assessment function may be regarded as the embodiment of the protocols and best practices of the organisation that is responsible for the working environment.

Therefore, the purpose of a classification and labelling system is three-fold: First, to facilitate unambiguous identification, to provide a controlled vocabulary of classification and labelling entities, including hazard classes and categories. Second, to facilitate unambiguous measurement, to define an ordering principle for specific classification and labelling entities, including hazard categories. Finally, to facilitate unambiguous assertion, to outline a framework for the specification of risk assessment functions, and the association of their codomain with specific chemical substances and mixtures.

3.2.2 Methodology for Modelling of CLP Regulation

In this section, we present our methodology for modelling the information content of the CLP Regulation, and our derived entity-relationship model.

As the source of the information content that we are working with is a legal document, our goal is to design a model that enables the RDF representation of said information content with the highest possible fidelity to the original text. Hence, we have selected RDFS as the modelling technology, rather than OWL, as it affords limited, but well-understood, semantics, and thus, minimises the risk of "overcomplicating" our model by providing more semantics than are necessary for the task. However, we would point out that our decision is the result of neither a failure of OWL nor a success of RDFS, but rather that it is a consequence of our "rote" approach to the modelling of legal documents, where entity and relationship are taken verbatim from the text.

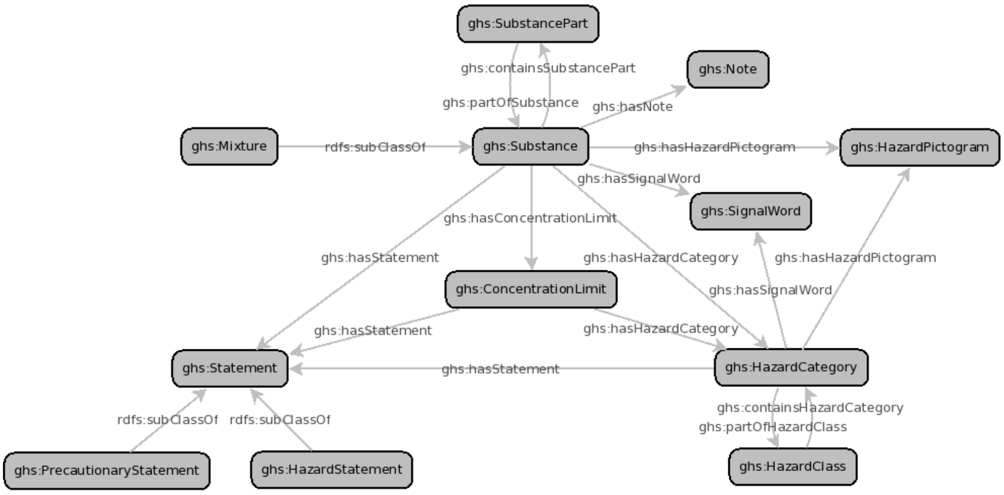


FIGURE 3.10: Depiction of RDF schema for core GHS entities and their inter-relationships (some entities and relationships are not shown).

In Figure 3.10, we give a depiction of the RDF schema for the core GHS entities and their inter-relationships. Entity names are taken verbatim from Article 2 of the CLP

Regulation, and are given in medial capitals (also known as "CamelCase"), e.g., the entity that denotes the concept of a "hazard class" is modelled as the `ghs:HazardClass` class. Relationship names are programmatically constructed according to the following naming scheme: The name of the codomain of the relationship is prefixed with a third-person, singular, present-tense verb, which indicates the nature of the relationship, e.g., the mereonomic relationship between a "hazard class" and its differentiations is modelled as the `ghs:containsHazardCategory` predicate. We now introduce the core entities of our model:

Hazard Class – a conceptual entity, which denotes a quality of a chemical substance; describes the nature of a physical, health or environmental hazard; represents the possibility that the phenomena associated with the hazard may be realised;

Hazard Category – a division of criteria within each hazard class, which specifies relative severity;

Hazard Pictogram – a graphical composition, which includes a symbol and other graphic elements, such as a background pattern or colour that is intended to convey the specific information about the hazard;

Signal Word – a word that indicates the relative severity of hazards, which may be used in order to alert readers to the presence of a potential hazard;

Statement – an abstract conceptual entity, which denotes a phrase;

Hazard Statement – a **Statement** that describes the nature of a hazard;

Precautionary Statement – a **Statement** that describes recommended measure(s) to minimise or prevent the adverse effects that are associated with exposure to a hazardous substance;

Substance – a conceptual entity, which denotes a chemical substance that is composed of exactly one "part";

Substance Part – a conceptual entity, which denotes a chemical element or compound in its natural state, or obtained via a manufacturing process;

Mixture – a **Substance** that is composed of two or more "parts";

Concentration Limit – a conceptual entity, which denotes a threshold of a classified impurity, additive or individual constituent "part" in a chemical substance, which may trigger the classification of the substance, with respect to a specific hazard; and

Note – a note relating to the identification, classification or labelling of chemical substances.

In our model, chemical substances are modelled as aggregations of one or more constituent "parts"[9]. There are two key advantages to this approach: First, and foremost, chemical information, such as chemical identifiers, may be associated with both the substance (as a whole) or with each individual part. Second, it is possible to differentiate between substances and mixtures (by simply counting the number of parts).

We include the `ghs:Mixture` class in our model for two key reasons: First, as we have discussed, while it is possible to differentiate between mixtures and substances, the computational cost of counting the number of parts for a substance is relatively high, especially when compared to the cost of asserting an additional `rdf:type` predicate. Second, it is useful for our model to capture the special semantics of the concept of a "mixture", e.g., a mixture is chemical substance, which is itself a set of substances, which have been mixed intentionally. Thus, to label an instance of the `ghs:Substance` class (of two or more `ghs:SubstancePart` instances) as an instance of the `ghs:Mixture` confers additional semantics.

Finally, in our model, chemical substances are indexed as follows:

Index Number – A numeric identifier, associated with an instance of the **Substance** class, of the form `ABC-DEF-GH-I`, where: `ABC` corresponds to the atomic number of the most characteristic element (or the most characteristic organic group, in the case of organic molecules); `DEF` denotes the consecutive number of the chemical substance in the series `ABC`; `GH` denotes the form in which the chemical substance is produced (or made available in the market); and `I` is a check-digit, which is calculated in accordance with the 10-digit ISBN method. All index numbers in annex VI of the CLP Regulation are guaranteed to be unique.

EC Number – A numeric identifier, associated with an instance of the **SubstancePart** class, of the form `ABC-DEF-G`, where: `ABC` and `DEF` are numbers; and `G` is a check digit, which is calculated using the 6-digit ISBN method. Also known as an EINECS, ELINCS or NLP number; it is the "official" (read: *de facto*) number of the chemical substance within the EU.

CAS Registry Number – A numeric identifier, associated with an instance of the **SubstancePart** class. CAS numbers are opaque, i.e., the syntax of the identifier has no inherent meaning. CAS numbers are guaranteed to be unique.

IUPAC Name – A textual identifier, associated with an instance of the **SubstancePart** class, which provides the name of a chemical substance according to the rules of the IUPAC nomenclature.

[9] As an interesting aside, the concept of a "substance of zero parts" (or the "null substance") is captured by the model, but it is not used, because, at least in theory, it should be represented as a singleton. In practice, it is also not very useful (and decidedly un-reactive!)

3.2.3 Methodology for RDF Representation of CLP Regulation

In this section, we present our methodology for the representation of the CLP Regulation as an RDF graph.

Prefix	Namespace URI	Description
ghs	http://ns.unece.org/ghs/	GHS vocabulary terms
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#	RDF vocabulary terms
skos	http://www.w3.org/2004/02/skos/core#	SKOS vocabulary terms
void	http://rdfs.org/ns/void#	VoID vocabulary terms

Table 3.3: Namespaces and prefixes used in Section 3.2.3.

We begin by creating a new namespace to demarcate RDF resources relating to the CLP Regulation, which is referred to as "ghs_id":

http://id.unece.org/ghs/

Next, we create a new URI to uniquely identify an RDF resource that describes the CLP Regulation itself, and may be used in order to assert bibliographic and provenance metadata:

http://id.unece.org/ghs/dataset

We model the dataset as an instance of `void:Dataset`. The key advantage of this approach is that the VoID vocabulary provides terms that, if asserted by data producers, allow data consumers to reason over the dataset itself, e.g., to infer the vocabularies that are used, the total number of triples, how many distinct classes and predicates are asserted, etc.

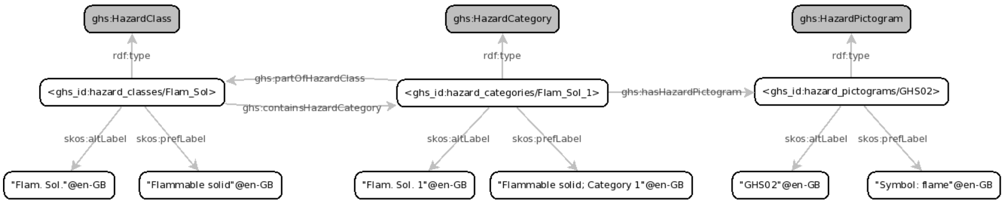


Figure 3.11: Depiction of RDF graph that describes the hazard category "Flammable solid; category 1", along with its associated hazard class and pictogram.

Next, we visit each page of annexes I-V, manually constructing a new RDF resource for every classification or labelling entity that is encountered. Lexical labels are associated with each entity using the `skos:prefLabel` and `skos:altLabel` predicates, which denote full and abbreviated names respectively. An example RDF resource, which describes the hazard category "Flammable solid; category 1", along with its associated hazard class and pictogram, is depicted in Figure 3.11.

The URI for the RDF resource that describes each classification or labelling entity is constructed by appending both the underscored and pluralized representation of the name of the class, and the URL-encoded representation of the British English translation of its “alt” label to the URI of the dataset:

```
http://id.unece.org/ghs/<Class>/<AltLabel>
```

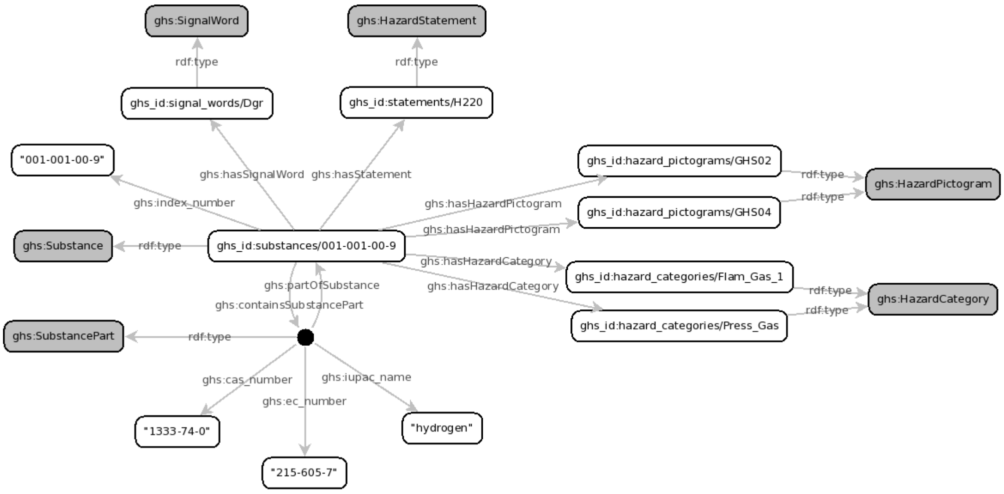


Figure 3.12: Depiction of RDF graph that describes the chemical substance “hydrogen”, along with its associated classification and labelling entities.

Finally, we visit each page of annex VI, manually constructing a new RDF resource for every chemical substance that is encountered. An example RDF resource, which describes the chemical substance “hydrogen”, along with its associated classification and labelling entities, is depicted in Figure 3.12.

The URI for the RDF resource that describes each chemical substance is constructed by appending the index number to the URI of instances of the **Substance** class within the dataset:

```
http://id.unece.org/ghs/substances/<IndexNumber>
```

In Figure 3.13, we give an exemplar SPARQL query that can be used in order to locate all instances of the **Substance** class, which are associated with an instance of the **SubstancePart** class, which itself has an IUPAC name that matches a specified regular expression, in this case, the case-insensitive string “aluminium”.

3.2.4 Summary of Web Interface

In this section, we present a web application that was specifically developed in order to present the information content of the CLP Regulation as a human- and machine-readable knowledge base.

```
PREFIX ghs: <http://ns.unece.org/ghs/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?substance ?substance_part ?iupac_name
WHERE {
  ?substance
    rdf:type ghs:Substance ;
    ghs:hasSubstancePart ?substance_part .

  ?substance_part
    rdf:type ghs:SubstancePart ;
    ghs:iupac_name ?iupac_name .

  FILTER regex(?iupac_name, "aluminium", "i")
}
```

Figure 3.13: Exemplar SPARQL query that locates all substances, whose IUPAC name contains the string “aluminium”.

The web application is available inside of the University of Southampton campus network at the following address:

http://ghs.chem.soton.ac.uk/

The web application was built using the Ruby on Rails (RoR) framework[10], which is itself based on the Ruby programming language[11]. RoR applications use the Model-View-Controller (MVC) architecture. The *model* is the domain of the application, i.e., the data. By default, the data for a RoR application is persisted using a relational database. The *view* is a representation of the model. Typically, applications use more than one view to construct their Graphical User Interface (GUI), e.g., by presenting the data in different ways. Finally, the *controller* mediates between the model and the views. Thus, the key advantages of using the MVC architecture is that it separates the concerns of each component of a large software application, encouraging developers to write modular and reusable code.

To generate the model for our web application, we map the RDF representation of the CLP Regulation to a relational database schema: RDFS classes are mapped to tables, datatype predicates are mapped to columns, and object predicates are mapped to foreign-key associations. Finally, individual RDF resources (instances of RDFS classes in our model) are mapped to records.

Our motivation for using a relational database, rather than a triple-store, accessed via a SPARQL end-point, is two-fold. First, while libraries for issuing requests to SPARQL end-points are available for the Ruby programming language, they are simply light-weight wrappers for the generic interface that is provided by all SPARQL end-points,

[10] http://rubyonrails.org/

[11] http://ruby-lang.org/

and hence, they do not support advanced functionalities, such as query optimisation and client-side caching. Furthermore, the development, testing and integration of these additional functions would be expensive (in terms of man-hours), and hence, would significantly delay the project. Second, the dataset is static. Thus, it is far simpler and more efficient to use a local copy, instead of repeatedly issuing queries to a remote data source.

A further benefit of using a relational database, or rather, using the "conventional" RoR approach, is that the web application is automatically afforded additional functionality. For example, based on the configuration of a RoR model, the framework automatically generates both JSON and XML-RPC representations, which may subsequently be used in order to construct RESTful web services.

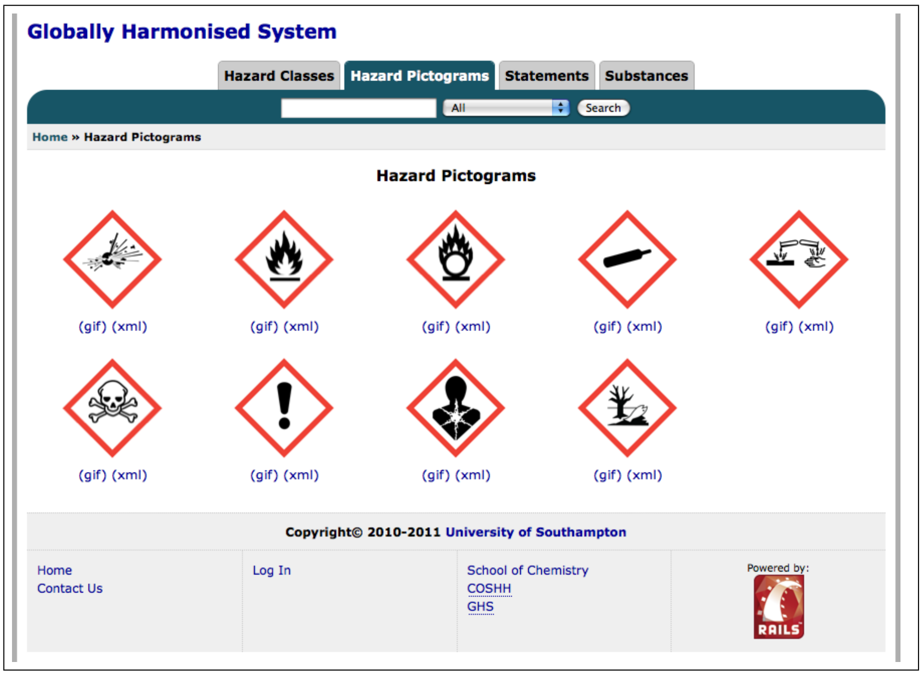
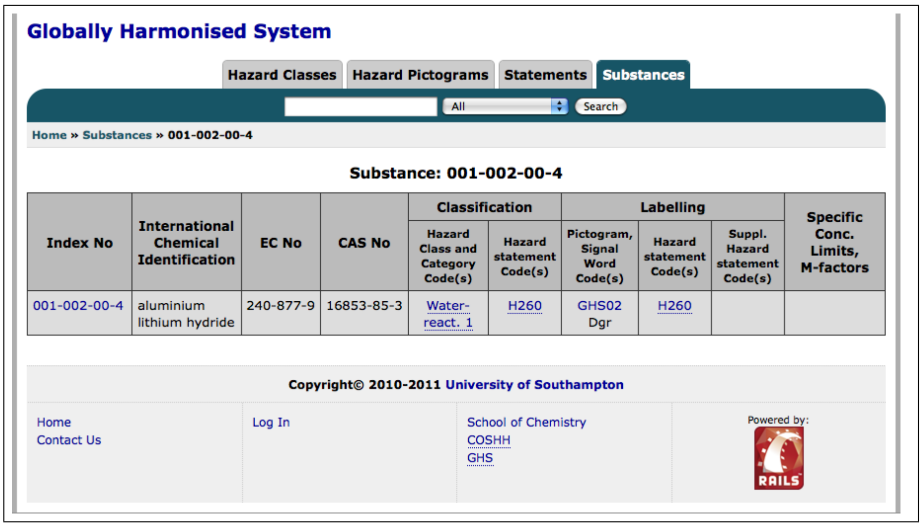


FIGURE 3.14: Screen shot of GHS web application; list of hazard pictograms (available at: http://ghs.chem.soton.ac.uk/hazard_pictograms).

In Figure 3.14, we present a screen shot of the GHS web application, which lists the nine instances of the **Hazard Pictogram** class that are described by the legislation. For each instance, the system provides both a human- and machine-processable description, along with a graphical depiction (as a bitmap image in GIF format).

In Figure 3.15, we present a screen shot of the GHS web application, which describes an instance of **Substance** class. To facilitate use of the web application by users who are familiar with the CLP Regulation, the user interface has been styled to resemble the table in Annex VI of the legislation. The fundamental difference between the web application and the [original] presentation of the legislation is that the labels for instances are rendered as hyperlinks.



Globally Harmonised System

Home » Substances » 001-002-00-4

Substance: 001-002-00-4

Index No	International Chemical Identification	EC No	CAS No	Classification		Labelling			Specific Conc. Limits, M-factors
				Hazard Class and Category Code(s)	Hazard statement Code(s)	Pictogram, Signal Word Code(s)	Hazard statement Code(s)	Suppl. Hazard statement Code(s)	
001-002-00-4	aluminium lithium hydride	240-877-9	16853-85-3	Water-react. 1	H260	GHS02 Dgr	H260		

Copyright© 2010-2011 University of Southampton

Home
Contact Us
Log In
School of Chemistry
COSHH
GHS

FIGURE 3.15: Screen shot of GHS web application; description of substance: "aluminium lithium hydride" (available at: ghs.chem.soton.ac.uk/substances/1).

3.2.5 Summary of Dataset

In this section, we present a summary of the dataset that was generated from the information content of the CLP Regulation.

Qualified Name	Frequency
ghs:ConcentrationLimit	453
ghs:HazardCategory	76
ghs:HazardClass	28
ghs:HazardPictogram	9
ghs:HazardStatement	97
ghs:Mixture	139
ghs:Note	19
ghs:PrecautionaryStatement	136
ghs:SignalWord	2
ghs:Substance	4136
ghs:SubstancePart	4522

TABLE 3.4: Instances of model entities in GHS dataset.

In Table 3.4, we give a summary of the number of instances of each class in our entity-relationship model (described in Section 3.2.2). Interestingly, 139/4136 instances of the **Substance** class were also found to be instances of the **Mixture** class. Thus, given the definitions of the two classes, and the dataset, we calculate the mean number of instances of the **SubstancePart** class per instance of the **Mixture** to be:

$$
\begin{aligned}
\frac{|\texttt{ghs:SubstancePart}| - (|\texttt{ghs:Substance}| - |\texttt{ghs:Mixture}|)}{|\texttt{ghs:Mixture}|} &= \frac{4522 - (4136 - 139)}{139} \\
&= \frac{525}{139} \\
&\approx 3.777\,(3\text{ d.p.})
\end{aligned}
$$

Finally, in total, the dataset describes an RDF graph of 109969 triples.

3.3 RSC ChemSpider

ChemSpider is an online chemical database[103], which was first launched in March 2007. In May 2009, ChemSpider was acquired by the Royal Society of Chemistry (RSC). At time of writing, the ChemSpider database contains descriptions of over 26 million unique compounds, which were extracted from over 400 third-party data sources. The core competencies of RSC ChemSpider are as follows:

Data integration (w.r.t. chemical information) – the ability to extract data about a single compound from multiple third-party sources, and aggregate it into a consistent whole;

Chemical identifier resolution – the ability to convert between chemical identifier formats, and to realise a chemical identifier as a ChemSpider record; and

Chemical structure and substructure search – the ability to search for compounds by providing a structure or substructure.

ChemSpider is a structure-centric database, which integrates data from multiple data sources. The database is populated by two distinct mechanisms: Web crawling and crowd-sourcing. Descriptions of new compounds are continuously located and downloaded from the Web by automated, unsupervised software applications called "crawlers", which autonomously walk the Web; follow hyperlinks; and download content. When new information is discovered by a crawler, the system attempts to locate a matching chemical structure in the database. If a match is found, then the new information is merged with that of the pre-existing record. Otherwise, a new record is created. Descriptions of new and existing compounds may also be created and/or modified by registered users.

In the ChemSpider database, a locally-unique identifier (referred to as the "RSC ChemSpider ID" or "CSID") is automatically associated with each record. The primary advantage of this approach is that, within the context of the database schema, there exists a one-to-many relationship between the CSID for a record, and the assertions that have been associated with that record (the systematic names, trade names, synonyms, registry numbers, chemical identifiers and descriptors, links to publications, etc). Consequentially, it is possible to distinguish between two records by analysing either their CSIDs or their respective sets of assertions. Furthermore, if it is subsequently discovered that two records describe the same compound, then one or both of the records can be deprecated, without affecting the rest of the database.

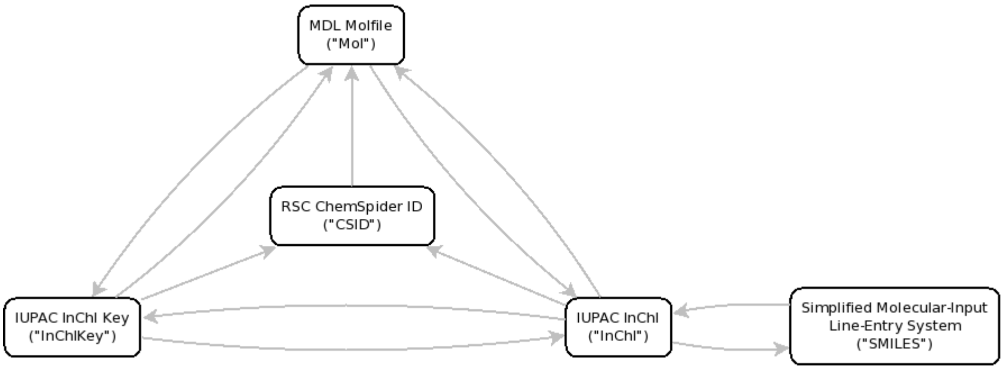

MDL Molfile ("Mol")
RSC ChemSpider ID ("CSID")
IUPAC InChI Key ("InChIKey")
IUPAC InChI ("InChI")
Simplified Molecular-Input Line-Entry System ("SMILES")

FIGURE 3.16: Depiction of the directed graph of "InChI" Web services provided by RSC ChemSpider, where nodes denote chemical identifier formats, and edges denote the availability of a Web service that provides an injective and non-surjective mapping for chemical identifiers from the source to the target format.

After a compound has been added to the database, and associated with a CSID, it is possible to search for said compound using any of its associated chemical identifiers. The relationship between a compound and a chemical identifier is inverse-functional, and hence, may be used to resolve said compound (using the chemical identifier). ChemSpider provides a suite of Web services for this purpose.

In Figure 3.16, we give the directed graph of Web services provided by RSC ChemSpider for chemical identifier resolution, where nodes denote chemical identifier formats, and edges denote the availability of a Web service that provides a mapping between the source and target formats. Note that the graph does not contain an edge from "MDL Molfile" to "RSC ChemSpider ID". This is because Molfiles[12] are used to express atomic connection tables, which describe the connectedness of sub-graphs of atoms, i.e., not a whole chemical structure, and hence, cannot be used as an inverse-functional identifier. In ChemSpider, Molfiles are used for chemical structure and substructure search, where multiple compounds may be returned as part of the result set.

Currently, the information content of the ChemSpider database is not available in a machine-processable format, and therefore, is non-trivial to incorporate into existing software applications. Thus, there is a need for the information content of the ChemSpider database to be made available in a machine-processable format, such as RDF. To

[12] http://www.mdl.com/solutions/

achieve this goal, we design and implement a new machine-processable, RDF representation for records in the ChemSpider database. We continue by implementing a suite of Web services, which leverage the new dataset.

The remainder of this section is organised as follows. First, we describe our methodology for modelling the information content of the RSC ChemSpider database, and present our functional and non-functional requirements for the dataset. Second, we describe our methodology for representing the information content of the RSC ChemSpider database as an RDF graph. Finally, conclusions are drawn regarding the new dataset.

3.3.1 Methodology for Modelling of RSC ChemSpider Records

In this section, we present our methodology for modelling the information content of the RSC ChemSpider database, and describe the functional and non-functional requirements for the dataset.

Due to the complexities of providing an alternate representation of the RSC ChemSpider database via a separately hosted Web service, it was decided that the system would be designed by the academics at the University of Southampton; and implemented and deployed by the software developers at the RSC. Thus, it was noticed during the initial stages that the success of the project would be dependent not on the correctness or technical completeness of the delivered product, but instead, on the quality of the communication between the two working groups during the design and implementation process. Ultimately, this reduced to the need for proper communication of the functional and non-functional requirements for the two most important aspects of the system:

Representation of Compounds – the methodology for the representation of an individual compound, and the relation of said compound to its associated information resources; and

Resolution of Compounds – the methodology for the identification and resolution of an individual compound by means of one or more of its associated chemical identifiers.

To model each record in the RSC ChemSpider database, it was decided that the ChemAxiom chemical ontology [104] would be used. This decision was motivated by three factors: the need for structure-centric modelling, correctness, and extensibility (of the dataset).

First, and foremost, it is crucial that our model corresponds to the high-level design of the RSC ChemSpider database, which, as alluded to earlier, is structure-centric, i.e., data about a single compound, retrieved from multiple sources, is integrated using chemical identifiers and structure descriptors. The ChemAxiom ontology is a good fit for this requirement, as it uses mereology (the theory of part-whole relations) in order to model a compound as a finite set of parts, where each part corresponds to a distinct moiety.

Second, it should be noted that, as we are not chemists (we are computer scientists), we do not possess the requisite skill-set for assessing the correctness or completeness of a given vocabulary. Instead, our methodology for selection is guided by our understanding of the provenance of each vocabulary (its authors and their prior art), and the applications of each vocabulary. Hence, we use the ChemAxiom ontology for the simple reason that it has been designed by chemists, and rigorously tested by computer scientists.

Third, it is important that new information can be added to the description of a compound at a later date, i.e., that our solution is extensible. ChemAxiom meets this requirement, as it uses reifications to represent factual assertions, rather than binary predicates. Each fact about a compound (or part of a compound) is stated via a typed entity, e.g., to associate a chemical identifier with a part of a compound, we construct a new resource, whose "value" is a chemical identifier, and whose type is a specific chemical identifier format. Thus, to implement a new chemical identifier format within the system, all that is necessary is to mint a unique identifier for said format.

In addition to describing a compound, each record in the RSC ChemSpider database is also associated with zero or more additional information resources, which include: two- and three-dimensional depictions of chemical structure; machine-processable representations of chemical structure; calculated and experimentally-determined chemical properties; spectra; and, hyperlinks to patents and journal articles.

To link additional information resources with the description of a compound, it was decided that the OAI-ORE standard [26] would be used. The key concept that is introduced by OAI-ORE is that of the "aggregation" (of Web resources). Each aggregation is identified by a single URI. Hence, when the URI for an aggregation is dereferenced, the consumer not only obtains a description of said aggregation, but also learns of the existence (and identity) of each aggregate resource. The primary advantage of this approach is that it provides a standardised mechanism for relating a record to its associated information resources.

Finally, it is important that our model incorporates information about the RSC, and the relationship between the data and the RSC. This is critical for two reasons. Firstly, the data must be appropriately licensed, otherwise it may not be reused by third parties. Second, it is remarkably valuable to co-promote the project, by associating it with the RSC brand, and by leveraging the high degree of trust that is placed in the brand by third parties, i.e., users of the system will be more likely to trust the data, if it is associated with a well-known brand.

3.3.2 Methodology for Representation of RSC ChemSpider Records

In this section, we present our methodology for the representation of RSC ChemSpider records as RDF graphs.

Prefix	Namespace URI	Description
chemdomain	http://www.polymerinformatics.com/ChemAxiom/ChemDomain.owl#	ChemAxiom ChemDomain Chemical Metadata vocabulary terms
dct	http://purl.org/dc/terms/	Dublin Core terms
foaf	http://xmlns.com/foaf/0.1/	FOAF vocabulary terms
ore	http://www.openarchives.org/ore/terms/	OAI-ORE vocabulary terms
owl	http://www.w3.org/2002/07/owl#	OWL vocabulary terms
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#	RDF vocabulary terms
void	http://rdfs.org/ns/void#	VoID vocabulary terms

Table 3.5: Namespaces and prefixes used in Section 3.3.2.

We begin by creating a new namespace to demarcate RDF resources, which is referred to as “chemspider”:

http://rdf.chemspider.com/

Next, we create a new URI to uniquely identify an RDF resource that describes the RSC ChemSpider dataset itself:

http://rdf.chemspider.com/void.rdf

We model the dataset as an instance of `void:Dataset`. The key advantage of this approach is that we may associate bibliographic and provenance metadata with the description of the dataset, e.g., we use the `dct:license` and `dct:publisher` predicates to assert the licensing and ownership for the contents of the dataset.

In Figure 3.17, we give a depiction of the RDF graph that describes a single compound from the ChemSpider database. In accordance with the ChemAxiom best principles [104]: each compound is modelled as an instance of `chemaxiom:NamedChemicalSpecies`; each moiety is modelled as an instance of `chemaxiom:MolecularEntity`; and, each chemical identifier is asserted as the reified codomain of the `chemaxiom:hasIdentifier` predicate. The key advantage of this approach is that, by decomposing each compound into its moieties, we necessarily construct multiple RDF resources, which can be used to assert additional relationships, and add new value to the data, e.g., by relating the

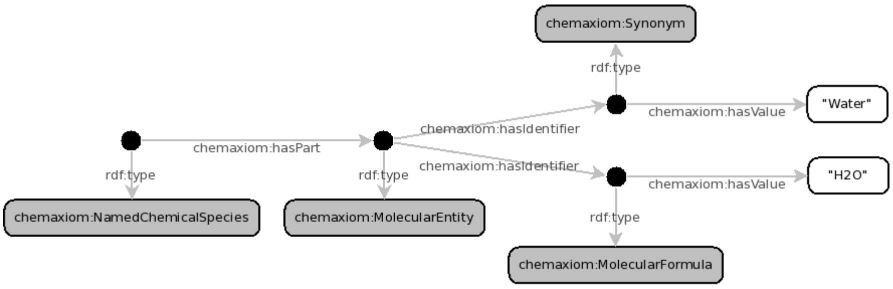


Figure 3.17: Depiction of RDF graph that describes the compound "Water" using terms from the ChemAxiom ontology (available at: http://www.chemspider.com/Chemical-Structure.937.rdf).

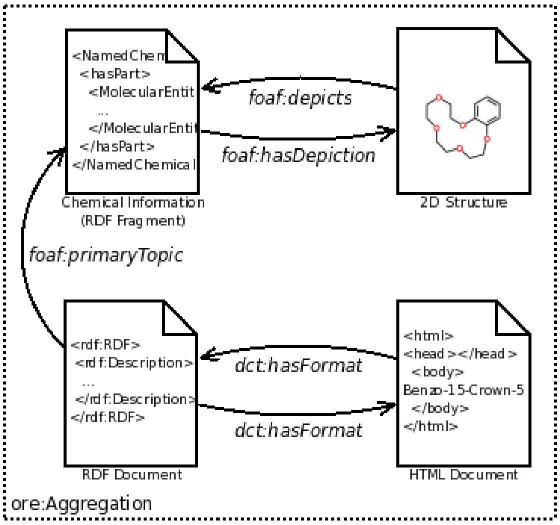


Figure 3.18: Depiction of OAI-ORE aggregation of information resources associated with an RSC ChemSpider record.

two- and three-dimensional depictions of a compound to its corresponding instance of `chemaxiom:MolecularEntity`.

In Figure 3.18, we give a depiction of an OAI-ORE aggregation of the information resources that are associated with a single record from the ChemSpider database, including: the human- and machine-readable representations of the record ("HTML Document" and "RDF Document"); the machine-readable description of the chemical substance ("Chemical Information"); and, the two-dimensional depiction of the chemical structure ("2D Structure"). The key advantage of this approach is that, by relating multiple resources as a single aggregation, which is identified by a URI, it is possible for users to discover all of the information that is associated with a ChemSpider record (and not just the information that constitutes the record itself). Moreover, the aggregates (the components of an aggregation) are formally demarcated from the rest of the ChemSpider database; providing a mechanism for subdividing the information content of the ChemSpider database into meaningful, macro-scale units.

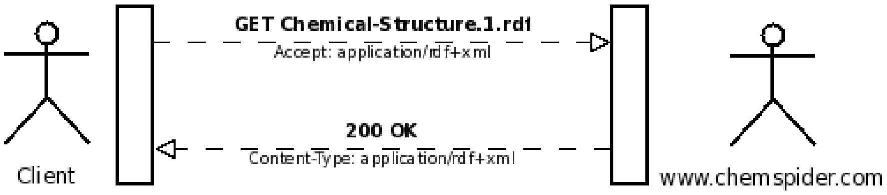

GET Chemical-Structure.1.rdf
Accept: application/rdf+xml
200 OK
Content-Type: application/rdf+xml
Client
www.chemspider.com

Figure 3.19: Activity diagram that describes the dereferencing of a URI, to give an OAI-ORE aggregation for an RSC ChemSpider record.

The aggregation for a compound is retrieved by dereferencing its URI, which is constructed from its CSID:

`http://www.chemspider.com/Chemical-Structure.<CSID>.rdf`

If the URI is valid, and a matching record is located in the database, then the server will generate a response with a "200 OK" status code (depicted in Figure 3.19). Otherwise, the server will respond with a "404 Not Found" status code.

Alternatively, a chemical identifier resolution service is provided, which may be used to retrieve the aggregation for a compound by resolving any of its chemical identifiers. Currently, the following chemical identifiers are supported by the service: InChI, InChIKey and CSID.

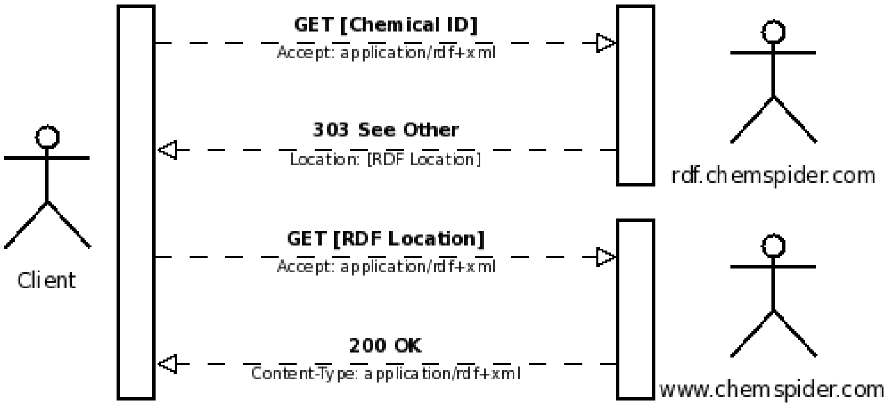


Figure 3.20: Activity diagram that describes the successful resolution of a chemical identifier, and dereferencing of the obtained URI, to give an OAI-ORE aggregation for an RSC ChemSpider record.

The URI for a chemical identifier is constructed by appending the URL-encoded representation of said chemical identifier ("ChemicalID") to the URI of the identifier resolution service end-point:

`http://rdf.chemspider.com/<ChemicalID>`

If the specified chemical identifier is valid, and a matching record is located in the database, then the server will generate a response with a "303 See Other" status code, whose "Location" header is the URI for the ChemSpider record (depicted in Figure 3.20).

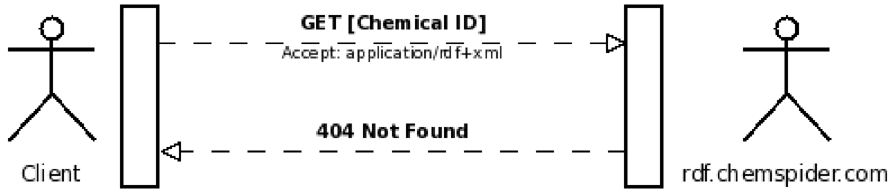


FIGURE 3.21: Activity diagram that describes the unsuccessful resolution of a chemical identifier.

Attempting to resolve an invalid chemical identifier, or the failure to locate a matching compound, will cause the server to generate an empty response with a "404 Not Found" status code (depicted in Figure 3.21).

There are two key advantages to this approach. First, URIs may be constructed programmatically (by the consumer), without searching the RSC ChemSpider database. Second, the mechanism provides a clear separation between the two main activities of resolving and dereferencing a chemical identifier. Before using the service, each user possesses a single piece of information: a chemical identifier. After the successful resolution of the chemical identifier, each user obtains two additional pieces of information: whether or not a corresponding record exists, and, if one does exist, the URI of said record. Hence, users of the service who only want to know about the existence of a record, but are not interested in its contents, do not need to dereference the resulting URI. Thus, the overall load on the ChemSpider software infrastructure is greatly reduced.

3.3.3 Summary of Dataset

In this section, we present a summary of the dataset that was generated from the information content of the RSC ChemSpider database.

The RDF interface to RSC ChemSpider was made publicly available in May 2011 [105]. Since publication, the dataset has grown substantially. The dataset now includes a machine-processable description of every record in the RSC ChemSpider database. At time of writing, this amounts to over 1.158×10^9 RDF triples. Moreover, since publication, the dataset has been used to integrate RSC ChemSpider itself with other online databases and web services, including: DBPedia and OpenMolecules.

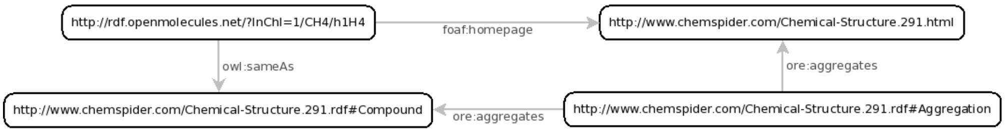

http://rdf.openmolecules.net/?InChI=1/CH4/h1H4
foaf:homepage
http://www.chemspider.com/Chemical-Structure.291.html
owl:sameAs
ore:aggregates
http://www.chemspider.com/Chemical-Structure.291.rdf#Compound
ore:aggregates
http://www.chemspider.com/Chemical-Structure.291.rdf#Aggregation

FIGURE 3.22: Depiction of RDF graph that asserts the relationships between the RSC ChemSpider and OpenMolecules descriptions of the compound "Methane" (available at: http://rdf.openmolecules.net/?InChI=1/CH4/h1H4).

In Figure 3.22, we give a depiction of an RDF graph that relates the RSC ChemSpider and OpenMolecules descriptions of the compound "Methane". The RDF graph describes and relates four resources: an OpenMolecules resource, an HTML document (provided by RSC ChemSpider), a description of a compound (provided by RSC ChemSpider), and an aggregation (provided by RSC ChemSpider). The relationships are interpreted as follows:

- The `owl:sameAs` predicate asserts that the OpenMolecules resource and the description of a compound are identical, i.e., that they have the same identity;
- The `foaf:homepage` predicate asserts that the HTML document provides the primary human-readable representation of the OpenMolecules resource; and
- The `ore:aggregates` predicates (included in the figure for completeness) assert that the description of a compound and the HTML document are aggregated by the aggregation.

The assertion of the `owl:sameAs` predicate by OpenMolecules is arguably correct. First, we note that OpenMolecules uses Chemical Information Ontology (CHEMINF), which is itself based on Semanticscience Integrated Ontology (SIO), to describe compounds, whereas RSC ChemSpider uses the ChemAxiom ontology. While the two ontologies address the same domain, they differ substantially in their scope and complexity. Hence, it may not be the case that the CHEMINF description asserts the same information as the ChemAxiom description of the same compound. However, the semantics of the `owl:sameAs` predicate are such that the two related resources are the same if and only if they have the same identity, i.e., that they share an inverse-functional identifier. Thus, the assertion is arguably correct as both resources share an InChI identifier.

The assertion of the `foaf:homepage` predicate by OpenMolecules is interesting, as it demonstrates the value of maintaining a human-readable representation of an information resource. OpenMolecules uses XSLT to transform its RDF descriptions into extremely basic HTML documents, which offer little additional value to data consumers. This is in contrast to RSC ChemSpider, which maintains a separate human-readable representation, which is related to the RDF description using OAI-ORE. The semantics of the `foaf:homepage` predicate are such that the codomain of the relationship is the primary human-readable document that describes the domain. Moreover, the semantics imply that the "homepage" is maintained by the "owner" of the resource in question. Since neither of these facts are the case, a suitable replacement would be the `foaf:page` predicate.

3.4 Control of Substances Hazardous to Health (COSHH)

In this section, we present a web service that uses our datasets in order to generate health and safety assessment forms.

The Control of Substances Hazardous to Health (COSHH) Regulations 2002 are a United Kingdom (UK) Statutory Instrument (SI) that governs the use of hazardous substances in the workplace in the UK [46]. Specifically, COSHH requires that employers provide information, instruction and training to any employees who will be exposed to hazardous substances.

One of the core aspects of COSHH is the requirement for conducting risk assessments. It is recommended that a risk assessment is conducted for each substance that is used in the workplace (regardless of whether or not the container for said substance is explicitly labelled as "hazardous").

To conduct a risk assessment for a substance, it is necessary to locate its classification, labelling and packaging information [106]. In the UK, the Chemicals (Hazard Information and Packaging for Supply) (CHIP) Regulations 2009 require that all suppliers provide this information in the form of a safety data sheet (SDS). Typically, the SDS is supplied as part of the packaging for a substance, or via the supplier's web site, however, many issues arise when this is not the case, and employees are required to manually locate and integrate the necessary information. These issues are compounded when we consider that it is common for procedures to involve more than one substance. What if substances are missed, and their hazards are not identified?

Clearly, many of these issues can be addressed with the application of computers. A potential solution would be the implementation of a health and safety assessment form generator that is informed by a knowledge base, where the knowledge base is an aggregation of machine-processable representations of the appropriate legislation. Thus, to generate a health and safety assessment form, all that would be required is to cross-reference a set of substances with the information in the knowledge base, and interpolate a template.

We proceed to implement the health and safety assessment form generator, using the methodology described above, as an extension of the GHS web application (described in Section 3.2.4). The service is available inside the University of Southampton campus network at the following address:

http://ghs.chem.soton.ac.uk/coshh/forms/new

To use the service, users supply a set of $\langle \textit{substance}, \textit{state}, \textit{quantity} \rangle$ triples, along with a plain-text description of the procedure. Each triple denotes one substance that will be used as part of a procedure, where: *substance* is a chemical identifier for the substance; *state* is the state of matter for the substance (given the environment in which the procedure will take place); and, *quantity* is the amount of the substance that will be used (in natural units). If the set contains more than one element, then it is assumed that the substances will be present in the same space at the same time. This assumption is necessary, as it is possible for certain pairs of substances to spontaneously react when their containers are in close proximity to each other.

In Figure 3.23, we present a screen shot of a health and safety assessment form that was generated from the GHS description of the substance "aluminium lithium hydride". To facilitate the use of the web service within the University of Southampton, the presentation of the health and safety assessment form has been styled using CSS to resemble the pre-existing, word-processor template.

3.4.1 Legal Implications of Deployment and Use of Automated Artefact Generation Service

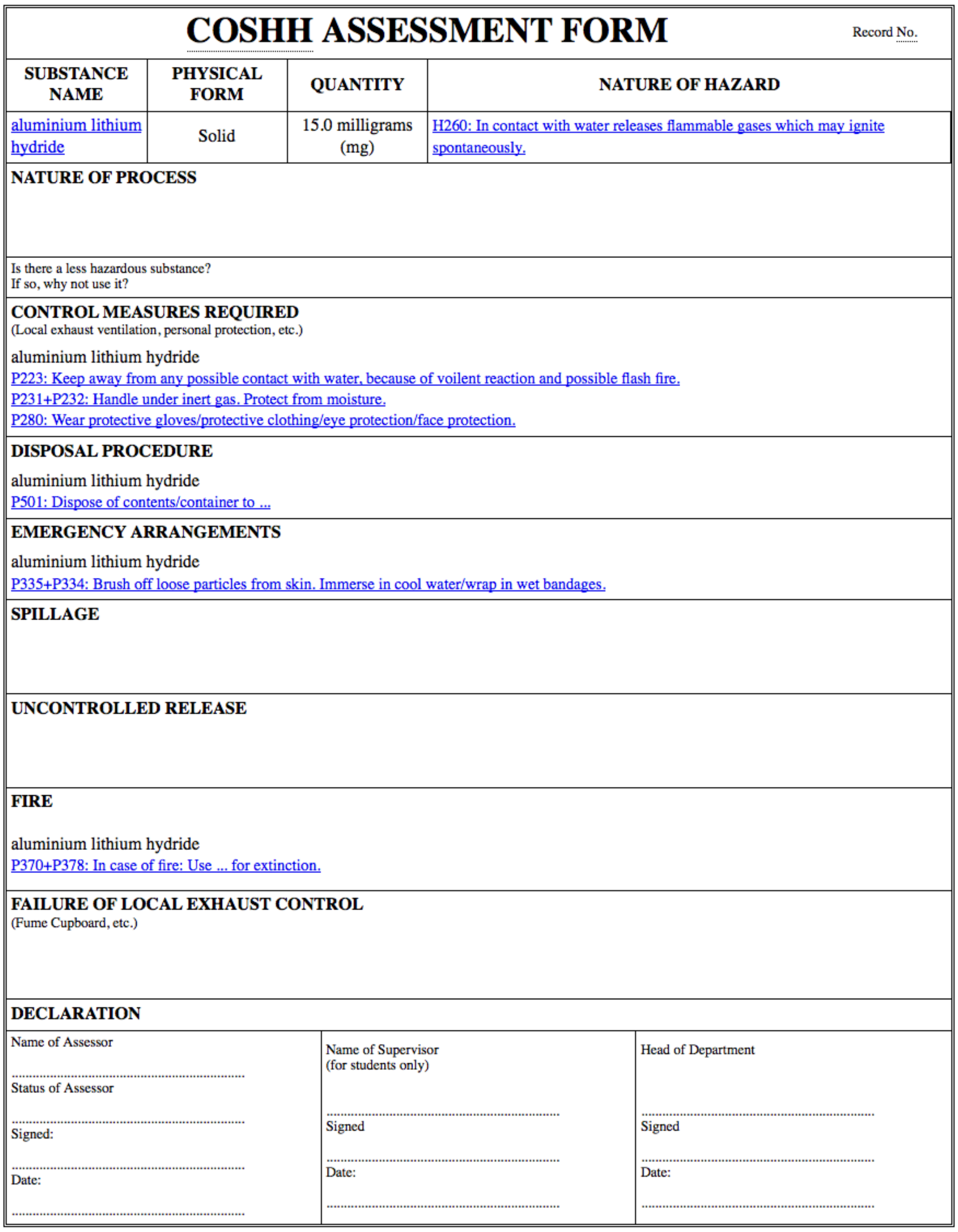
Following the deployment of the service, issues were raised about the legal implications of the deployment and utilisation of an automated health and safety assessment form generator. The issues can be summarised as follows:

Validity – In order to perform a health and safety assessment, it is necessary to construct a formal description of the procedure (that will be enacted in the future). Given the description of the procedure, it is possible to enumerate the set of substances (that will be used in the future). Given the set of substances, it is possible to enumerate the set of classification and labelling entities (that are relevant to the given substances). Thus, if we assume that both the initial description of the procedure and the subsequently applied mechanisms are valid, then is it correct to infer that the result (a completed health and safety assessment form) is valid?

Accountability – Regardless of the validity of the description of the procedure, who has legal blame in the event that the information that is asserted by a health and safety assessment is incorrect: the third-party, who provided the information; the organisation, who sanctioned the use of the third-party service; or, the individual, who accepted the validity of the information?

Value Proposition – Is the net utility that is obtained by the individual, when he manually performs a health and safety assessment, greater than the net utility that is obtained by the organisation, when it delegates the performance of health and safety assessments to a third-party service provider?

These issues are particularly interesting, not only because of their legal (and philosophical) implications, but also because they can be generalised in order to describe



COSHH ASSESSMENT FORM Record No.

SUBSTANCE NAME	PHYSICAL FORM	QUANTITY	NATURE OF HAZARD
aluminium lithium hydride	Solid	15.0 milligrams (mg)	H260: In contact with water releases flammable gases which may ignite spontaneously.

NATURE OF PROCESS

Is there a less hazardous substance?
If so, why not use it?

CONTROL MEASURES REQUIRED
(Local exhaust ventilation, personal protection, etc.)

aluminium lithium hydride
P223: Keep away from any possible contact with water, because of voilent reaction and possible flash fire.
P231+P232: Handle under inert gas. Protect from moisture.
P280: Wear protective gloves/protective clothing/eye protection/face protection.

DISPOSAL PROCEDURE

aluminium lithium hydride
P501: Dispose of contents/container to ...

EMERGENCY ARRANGEMENTS

aluminium lithium hydride
P335+P334: Brush off loose particles from skin. Immerse in cool water/wrap in wet bandages.

SPILLAGE

UNCONTROLLED RELEASE

FIRE

aluminium lithium hydride
P370+P378: In case of fire: Use ... for extinction.

FAILURE OF LOCAL EXHAUST CONTROL
(Fume Cupboard, etc.)

DECLARATION

Name of Assessor .... Status of Assessor .... Signed: .... Date:	Name of Supervisor (for students only) .... Signed .... Date:	Head of Department .... Signed .... Date: ....

Figure 3.23: Screen shot of COSHH assessment form generated from GHS description of substance: "aluminium lithium hydride" (index number: 001-002-00-4).

any artefacts that were procedurally-generated using the information content of a finite knowledge-base:

Validity* – If we assume that both the procedure and its inputs are valid, then is the resulting procedurally-generated artefact valid?

Accountability* – Who has legal blame for the consequences of trusting the information content of a procedurally-generated artefact?

Value Proposition* – Is the net utility that is obtained by the individual, when he manually generates an artefact, greater than the net utility that is obtained by the organisation, when it delegates the act of artefact generation to a third-party service provider?

Clearly, the issue of "validity" is deeply important, e.g., within the context of a laboratory environment, the acceptance of, and subsequent reliance on, an "invalid" health and safety assessment could have negative consequences for all involved. Hence, it is natural to ask to the question: when is a procedurally-generated artefact "valid"?

To provide an answer, we must consider the semantics of the adjective "valid", and its inverse "invalid". Thus, the concept of the "validity" of a procedurally-generated artefact is defined as follows: A procedurally-generated artefact is "valid" if and only if both its constituents and its generator (the procedure that generated the artefact) are themselves "valid", otherwise, it is "invalid"[13].

Given our definition, it is clear that from the point of view of an individual, who is employed by an organisation, the "validity" of a procedurally-generated artefact must be taken on faith, based on the assumptions that (a) their employer has sanctioned the use of a "valid" third-party service; and, (b) that they are providing "valid" inputs for said service. Similarly, from the point of view of an organisation, the "validity" of a procedurally-generated artefact must also be taken on faith, with the assumptions that (c) the third-party is providing a "valid" service; and, (d) that their employees are providing "valid" inputs for the service.

Clearly, there are symmetries between assumptions (a) and (c), and assumptions (b) and (d). The symmetry between assumptions (a) and (c) encodes an expectation that is held by the individual about the past actions of the organisation, which may or may not be backed-up by an explicit assertion of truth, i.e., the individual assumes that their employer has sanctioned the use of the third-party service, because he has been asked to use it. Similarly, the symmetry between assumptions (b) and (d) encodes an expectation that is held by the organisation about the future actions of the individual, which may

[13] Interestingly, with this definition of "validity", it is not only acceptable, but also quite practical, to consider the generator itself as a constituent of a procedurally-generated artefact, i.e., that the generator is evaluated via a higher-order mathematical function – a general-purpose generator evaluator.

or may not be backed-up by an explicit assertion of truth, i.e., the organisation assumes that the individual will use the service consistently and correctly.

Therefore, in the event that any party (the individual, organisation, or service provider) has reason to believe that any of the offerings of any of the other parties are "invalid", then these assumptions are manifest as statements of accountability, responsibility, and, ultimately, legal blame. These statements are summarised as follows:

- An individual is accountable if he provides an "invalid" constituent for a procedurally-generated artefact;
- An organisation is accountable if it sanctions the use of an "invalid" third-party service;
- A third-party is accountable if it provides an "invalid" service.

Clearly, the truth of these statements could be determined if all of the parties that are involved agree to assert the provenance of their offerings. However, it is important that we consider both the positive and negative effects of the resulting sharp increase in the level of transparency. To paraphrase the character Benjamin "Uncle Ben" Parker from the Spider-Man comic books[14]: with great [organisational] transparency comes great [individual] accountability, i.e., within the context of a provenance-aware system, if an event occurs, and the system can identify its effects, then the system can usually identify its causes (or said differently: within the context of a provenance-aware system, there is almost always someone to blame).

In this case, as we are the third-party, to limit our legal responsibility, the service was modified to take the following precautionary measures:

- Source code and datasets are publicly accessible;
- Assessment forms, and the data from which they were generated, are not persisted;
- Templates include a formal declaration, which must be signed and countersigned as part of the assessment form approval procedure.

3.4.2 Value Proposition for Deployment and Use of Automated Artefact Generation Service

To gain a broader understanding of the third issue that was described in Section 3.4.1, a cost-benefit analysis for the deployment and utilisation of an automated artefact generation service was conducted from the perspective of the three parties: the individual, the organisation, and the service provider.

[14] http://en.wikipedia.org/wiki/Uncle_Ben

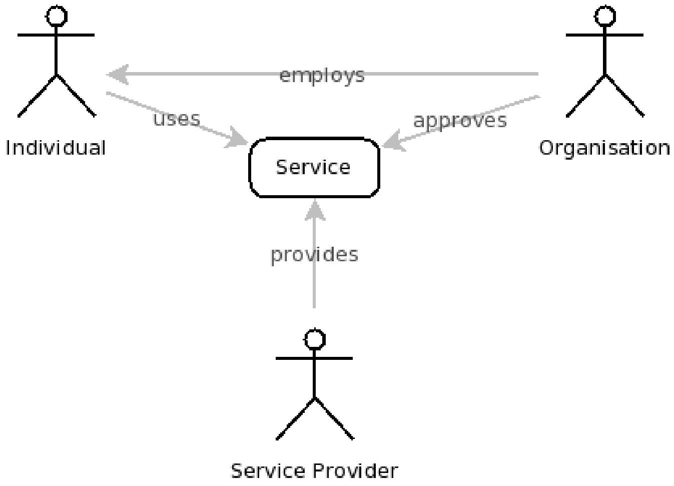


Figure 3.24: Depiction of the relationships between the three considered parties (the individual, organisation, and service provider), and the service that is being provided.

In Figure 3.24, we present a depiction of the relationships between the three considered parties. The relationships are summarised as follows:

- The service provider "provides" the service;
- The organisation "approves" the use of the service;
- The organisation "employs" the individual; and,
- The individual "uses" the service.

3.4.2.1 Value Proposition for Individual

From the perspective of an individual (who is employed by an organisation), the benefits of using an automated artefact generation service are that working time will be used more efficiently, and that both the format and information content of artefacts are standardised. Generally, the working time of an employee can be divided into two phases, artefact generation and artefact utilisation. If the process of generating artefacts is automated, then the proportion of time spent generating artefacts should be reduced, and, consequentially, a larger proportion of working time will be spent using the generated artefacts. Furthermore, automation of the artefact generation procedure ensures that both the structure and semantics of artefacts are consistent.

In contrast, from the perspective of an individual, the drawbacks of using an automated artefact generation service are an increase in the perceived level of accountability and personal liability, and, due to the automation, a reduction in the number of opportunities to learn about the artefact generation procedure. As we have discussed, when an automated artefact generation service is deployed, the individual becomes responsible for supplying "valid" inputs and, therefore, accountable for supplying "invalid" inputs. Hence, from the perspective of some individuals, the deployment and subsequent usage of an automated artefact generation service could expedite the discovery of their personal failures, which, for better or worse, may result in the termination of their employment. Moreover, if the deployment of the service is successful, then, given the rate of natural wastage at their employer, there is a risk that new employees will not learn, and existing employees will not practice, the skills that are necessary to manually perform the artefact generation procedure. Hence, there is a significant risk that, in the event that the automated service is unavailable, no artefacts will be generated, and, therefore, no artefacts will be utilised, i.e., if the artefacts are critical to the task, which is certainly the case for health and safety assessments, then the consequence is that absolutely no work can be performed, and hence, the employer incurs a loss.

3.4.2.2 Value Proposition for Organisation

From the perspective of the organisation (that employs individuals), the benefits of deploying an automated artefact generation service mirror those of the individual, e.g., that employee working time will be used more efficiently, and hence, that employees will be more productive; and, that both the format and information content of artefacts are standardised across the organisation.

However, from the perspective of the organisation, the drawbacks of deploying an automated artefact generation service are both numerous and varied. For example, notwithstanding the immediate costs of service deployment and maintenance[15], and employee training, organisations also incur a continuous cost in order to mitigate the risk of employees generating "invalid" artefacts, e.g., by employing additional personnel to act as supervisors. Moreover, from the perspective of the organisation, there is a critical disadvantage to the deployment of an automated artefact generation service, which is provided by a third-party: information leakage. If the service is deployed in a central location, which is outside of the organisational boundary, then information must necessary traverse said boundary in order for artefacts to be generated. Thus, there is a significant risk that if the service (or service provider) is compromised, then proprietary information may be intercepted, stolen, and used to benefit the organisation's competitors.

3.4.2.3 Value Proposition for Service Provider

Finally, from the perspective of the service provider, the benefits of an organisation's decision to deploy their automated artefact generation service are obvious. Firstly, there

[15] It should be noted that for a centralised, Web-based service, such as an automated health and safety assessment form generator, from the perspective of the organisation, the cost of deployment is essentially zero, as the service does not require the organisation to provision any internal infrastructure (with the notable exception of Internet connectivity).

is the immediate incentive of financial remuneration for the service provider, e.g., for a fee, the provider may license the use of the service by the organisation. However, in the event that the service is provided for no cost, then a second benefit may be derived: deployment of the service within the organisation creates new opportunities for brand association and co-promotion, i.e., the service provider may benefit from the assertion that "organisation X is using our product".

Similarly, from the perspective of the service provider, the drawbacks of deploying an automated artefact generation service are both obvious. Firstly, there is the immediate, and unavoidable, cost of the software development process, which not only includes the cost of the generation of the source code for the software itself, but also the cost of the generation of the required datasets. Secondly, as we have discussed, the service provider must also mitigate against the risk of the service generating "invalid" artefacts.

3.4.2.4 Summary of Value Proposition

In this section, we have presented a cost-benefit analysis for the deployment and utilisation of an automated artefact generation service, from the perspective of three parties: an individual, an organisation, and a third-party service provider.

A summary of the cost-benefit analysis is given in Table 3.6. Given our analysis, we draw the following conclusions:

- From the perspective of an individual (who is employed by an organisation), the costs significantly outweigh the benefits, due to the perception of increased personal liability and legal accountability;
- From the perspective of an organisation (who employs individuals), the benefits are balanced by the costs, i.e., while the deployment of the service may improve efficiency and productivity, there are also significant risks associated with the use of an automated service;
- From the perspective of a service provider, the benefits of financial and marketing opportunities clearly outweigh the costs of development and maintenance.

	Individual	**Organisation**	**Service Provider**

Cost(s)	Increased accountability;
	Risk of generating (and/or using) an "invalid" artefact;
	No opportunity to learn (and/or practice) manual artefact generation procedure.
		Cost of deployment and maintenance;
		Cost of employee training;
		Risk of employees generating (and/or using) "invalid" artefacts;
		Risk of employees not learning (and/or practicing) manual artefact generation procedure;
		Risk of employees relying on automated services;
		Risk of information leakage.
			Cost of development and testing;
			Risk of providing an "invalid" service.

Benefit(s)	Increased efficiency and productivity; Quality assurance.	Increased efficiency and productivity; Quality assurance.	Financial incentives (remuneration); Opportunities for marketing, branding and co-promotion.

Table 3.6: Cost-benefit analysis for the deployment and utilisation of an automated artefact generation service, e.g., a health and safety assessment form generator.

3.5 Summary

In this chapter, we have presented three new datasets, where each dataset is a machine-processable representation of a pre-existing human-readable information resource, and a software application, which uses the new datasets, in order to provide a much-needed service to laboratory-based researchers.

We have constructed a machine-processable representation of the subject index of the third edition of the IUPAC Green Book as a controlled vocabulary, and, after analysis, have concluded that the original data (the subject index) is well-constructed, with excellent page coverage and relevance. Furthermore, we have found that the availability of a machine-processable controlled vocabulary would be very useful for researchers, as it would provide them with a consistent set of keywords that could be used in their publications. However, we have noticed that researchers may be reluctant to use a controlled vocabulary, unless it is derived from an authoritative text, or associated with a trusted brand.

We have constructed a machine-processable representation of the information content of the CLP Regulation. The new dataset identifies and describes the classification, labelling and packaging entities that are specified by the regulation. The dataset is also highly extensible, and has been used in order to describe over four thousand potentially-hazardous chemical substances and mixtures. We have found the availability of a machine-processable representation of the CLP Regulation would be highly valued by laboratory-based researchers, as it would enable them to construct high-quality health and safety assessments, where chemical substances are unambiguously identified, and automatically related to their specific hazards.

In collaboration with the Royal Society of Chemistry (RSC), we have enhanced the ChemSpider online chemical database by providing a machine-processable representation of all records via a Linked Data interface. The goal of the collaboration was to demonstrate that providing a machine-processable of existing data would be both inline with the core competencies of RSC ChemSpider (data integration, unambiguous identification of chemical structures, and structure-based search), and provide new value to RSC ChemSpider users. We devised a methodology for the representation of RSC ChemSpider records as machine-processable data, which was successfully applied to every record in the database. As testament to the success of our approach, we have subsequently discovered that both DBPedia and OpenMolecules have integrated their datasets with that of RSC ChemSpider.

Finally, we have demonstrated the reuse and of the new datasets by developing an automated, legally-compliant health and safety assessment form generator. To use the service, users specify a set of tuples, where each tuple describes an individual chemical substance that will be used as part of an experiment. The chemical substances are referenced using the same identifiers as CLP Regulation and RSC ChemSpider datasets. Thus, assessment forms that are generated by the service can be integrated into other software applications. To complement this work, we conducted a cost-benefit analysis for the deployment of an automated health and safety assessment form generator within an organisation that employs individuals in order to perform experiments. We concluded that automated artefact generation services (in this case, the "artefacts" are health and safety assessment forms) are disruptive technologies with multi-faceted value propositions. We found that individuals are likely to be against the deployment of such an automated service, as they believe that it would invert the direction of accountability within the organisation, and thus, increase their personal level of legal responsibility. Conversely, we found that organisations are likely to accept the deployment of automated services, as they believe that it will increase organisational transparency and employee efficiency, and improve the quality of any subsequently generated artefacts. Moreover, we found that service providers are likely to continue developing, deploying and maintaining these services, as they provide opportunities for brand association and co-promotion.

Throughout this chapter, we have argued for the development of machine-processable datasets and automated systems, which can be used by laboratory-based researchers in order to enhance their workflows, and add new value to their artefacts and publications. However, we have also identified the many risks that arise from the naïve reuse of these datasets and services, where the truth (or falsity) of assertions and validity of artefacts are blindly accepted without proof or explanation, e.g., that an automatically-generated health and safety assessment form is "valid" because it has been generated by an "approved" third-party service. Clearly, the most sustainable mechanism for the mitigation of these risks is the formal exposition of provenance. Thus, in the next chapter, we describe a vocabulary and methodology for the exposition of both prospective and retrospective provenance of formal processes.

Chapter 4

A Provenance Model for Scientific Experiments

In the previous chapter, we noted that one of the main barriers to the utilisation of machine-processable datasets and automation in the laboratory is the limited availability of provenance information. Without provenance information, it is impossible for researchers to establish the truth (or falsity) of the assertions of said datasets, or the validity of the actions that are performed, and the artefacts that are generated, by automated systems. By communicating the provenance of their offerings, data providers empower consumers to make informed decisions about trust.

In this chapter, we begin to address this issue by introducing an ontology for the exposition of both prospective and retrospective provenance of formal processes, which are enacted both *in silico* and *in vivo*. The ontology is informed by a philosophical consideration of the nature of formal processes, which is conducted within a reductionist framework, and informed by a set of principles. To evaluate the application of our approach for specific domains of discourse, we demonstrate the specialisation of the ontology for the description of a crystallography workflow – crystal structure determination.

The goal of this work is to outline a program for the implementation of a provenance-aware space, such as a laboratory, i.e., an environment, where provenance information can be captured for all activities that are performed therein.

The contributions of this chapter are as follows:

1. A philosophical consideration of the nature of formal processes;
2. An ontology for the exposition of both the prospective and retrospective provenance of formal processes;
3. A meta-process for the enactment of any other formal process (that is described in terms of the ontology); and

4. A Linked Data interface for the eCrystals repository for crystal structures; and

The remainder of this chapter is organised as follows. First, we present a philosophical consideration of the nature of formal processes. Second, an ontology for the exposition of the provenance of formal processes is presented. Third, a meta-process, whose enactment constitutes the enactment of any other formal process is presented. This is followed by the presentation of a Linked Data interface for the eCrystals repository for crystal structures. Finally, conclusions are drawn.

4.1 Reflections on Formal Processes

In this section, we present a philosophical consideration of the nature of formal processes. This work is conducted within a reductionist framework, i.e., we attempt to understand the nature of formal processes by a process of decomposition, where the concept of a formal process is treated as a complex component, which is itself built from less complex components.

The remainder of this section is organised as follows. First, we define the concept of a formal process. Second, a discussion of the act of description is presented. This is followed by a discussion of the act of description of a formal process. After that, we describe how to distinguish between the endurants of our system. Finally, conclusions are drawn.

4.1.1 Definition of a Formal Process

A "process" is a sequence of actions, where each action corresponds to an event, which affects the state of its environment.

A "formal process" is a description of a process, realised as an information resource.

4.1.2 The Act of Description

To describe something is to perceive it, and to make assertions of subsequent observations and measurements, or, in other words, to generate data. This data is synthesised into instances of abstract models, which are encoded as data structures, and interpreted according to specialised semantics.

When designing a model, the designer must fix a frame of reference – a perspective, from which the observers of the system may make their observations and measurements. However, by fixing the frame of reference, the space of possible assertions is necessarily restricted, as it is no longer possible to make observations from other frames of reference. Hence, the designer imposes their subjective interpretation on the domain of discourse, which is subsequently forced upon users.

Consider the dimension of time. According to the classical laws of physics, time offers the observer (who is fixed in the present) three points to cast his gaze: the past, present and future. Thus, to completely describe an event, which occurs at a precise point in time, it is necessary to make assertions from three distinct perspectives:

Prospective – Intentions and expectations.

Present – Sense perceptions.

Retrospective – Observations and measurements.

For example, consider the following sequence of events:

> A man is at home with his wife, preparing to go to the park, to meet his friends, and watch a cricket match. While explaining the history of the sport to his wife, the man states that he expects that, in accordance with tradition, "the ball will be red". The man leaves his home, and makes his way towards to the park. The match begins, and the man observes that "the ball is luminous pink". The match ends, and the man returns home. He notes in his journal and recounts to his wife that, to his great surprise, "the ball was luminous pink". Very surprising indeed!

Notice that, in order to completely describe the events that took place – the measurement of the colour of the cricket ball – it was necessary to invoke all three temporal perspectives. The intention of the man was to measure the colour of the ball. This was informed by the expectation that the colour of the ball would be red. The expectation itself was informed by the man's knowledge of the traditions of the game.

During the match, photons were scattered by the surface of the ball. Some of the photons entered the man's eye, and were detected by cells on the surface of his retina. The interactions were translated into electrical signals, which traveled to the man's brain. After the match, the man persisted the measurement of the colour of the ball in his journal. Finally, the measurement was annotated in a way that denoted the emotion of surprise.

4.1.3 Description of a Formal Process

The prospective description of a formal process is an assertion of the intentions of the observer; the sequence of actions that may be enacted (at an indeterminate point in the future), and the observer's expectations about the consequences of these actions, i.e., the predicted effect of each action on the state of the environment. In contrast, the retrospective description of a formal process is an assertion of the work done by the observer; the sequence of actions that were enacted (at a precise point in the past), and the observer's measurements of the effects of each action on the state of the environment. Given an observable phenomenon, there is a finite duration of time between the realisation of said phenomenon, the act of observation, and the persistence of any measurement data. Thus, the present description of a formal process is undefined, as all measurements must necessarily be asserted in the past tense, i.e., retrospectively.

In his writings [107, 108], Popper theorises that, at its core, the scientific method is an iterative, formal process, where participants are driven by empiricism, in a never-ending search for the "best" explanations. While we respect this view, we also agree with Maxwell's critique [109], which argues that, as the value of an explanation (in isolation) is incomparable, the burden of proof falls on the participants to explain their explanations, i.e., to describe the processes from which their explanations are derived. Only when this supplementary information is provided, Maxwell argues, can the value of competing explanations be compared. In this thesis, we argue that the enactment of a formal process is intentional [110, pp. 9–13], and hence, that the representation of the description of a formal process is both syntactic and formal, i.e., we assume that both Popper's "explanations" and Maxwell's "explanations of explanations" are representable.

In his essays [111, pp. 3–4, pp. 83–102], Davidson argues that such higher-order explanations are in fact rationalisations; specialised causal relations between an agent's intentions and their actions, and hence, posit "an agent's reasons for doing what he did". In contrast to other authors [112, 113], who argue that such a simplistic position should be abandoned, we hold Davidson's argument to be correct. Accordingly, we assume that the relationships between Popper's "explanations" and Maxwell's "explanations of explanations" are Davidson's "rationalisations".

The prospective and retrospective descriptions of a formal process are related by the concept of "actualisation" (also referred to as "realisation") – the act of making real. This concept is codified by a binary predicate, "is an actualisation of," which relates a retrospective description (the domain) to a prospective description (the codomain). The predicate is defined as follows: the entity that is described by the domain is an actualisation of the entity that is described by the codomain. Our definition of this relationship leverages the granular nature of the events that are being described. At the macro scale, the retrospective description is an actualisation of the prospective description, i.e., the retrospective description is a record of the enactment of the sequence of actions that is formalised by the prospective description. Similarly, at the micro scale, each action (that is part of the retrospective description) is an actualisation of a corresponding action (that is part of the prospective description).

Asserting the relationship between the prospective and retrospective descriptions of a formal process is beneficial for two key reasons. First, and foremost, the assertions make the context for each retrospective description explicit. Without an assertion of actualisation, the context for a retrospective description is purely existential, i.e., events transpired "because they did". Second, assertions of actualisation may be used as the logical building blocks for deriving more complex properties of retrospective descriptions of formal processes.

An example of such a property is "satisfaction", which codifies the notion of the fulfilment of one's expectations, and may be expressed in English as follows:

> A prospective description P is satisfied by a retrospective description R, if and only if, for each action p in P, there exists an actualisation r in R.

More formally, the property is expressed using first-order logic as follows:

$$\forall P\ \exists R\ \forall p\, ((p \in P) \wedge \exists r\, ((r \in R) \wedge \text{isActualisationOf}\,(r, p))) \rightarrow \text{isSatisfiedBy}\,(P, R)$$

We make the following observations:

- The (second) universal quantification holds for all elements of the set of actions in the prospective description, and requires the existence of an actualisation in the retrospective description. Consequentially, it possible for one to exceed one's own expectations, by deviating from one's original intentions, and performing additional actions, or describing additional artefacts.
- The definition invokes the micro scale (of individual actions), rather than the macro scale (of whole processes). Thus, a single retrospective description may satisfy multiple prospective descriptions simultaneously.
- No restriction is placed on the number of times that an individual action may be actualised. Hence, any action may be repeated, any number of times, where each repetition creates a distinct chain of events.

Given the context of a prospective description, the assertion of satisfaction does not imply that the retrospective description is finished. In contrast, the assertion of satisfaction simply implies that the retrospective description is able to finish. Hence, once a retrospective description has satisfied its counterpart(s), the observers of the system may decide to either: do more work; or, to stop, perform any bookkeeping or administration tasks, and assert the retrospective assertion as being "finished". There are two key benefits to this approach. First, what was previously an implicit action is lifted to being an explicit action, which may be modelled like any other component of the system.

Second, the state of the retrospective description is made explicit. Put simply, one may ask the system: "which enactments are finished?"

In summary, to fully describe a formal process, it is necessary to invoke both a prospective and retrospective frame of reference, and to assert both our intentions and our actions, and to relate those assertions using the concept of actualisation. In this way, it is possible not only to get satisfaction, but also to deviate from one's original intentions, and exceed one's expectations, without compromising the consistency of our assertions.

4.1.4 To Distinguish Between Observers and Artefacts

In order to describe a formal process, we have identified at least two classes of continuant: observers and artefacts. In this section, we ask the question: What distinguishes an observer (of a system) from an artefact (of the same system)?

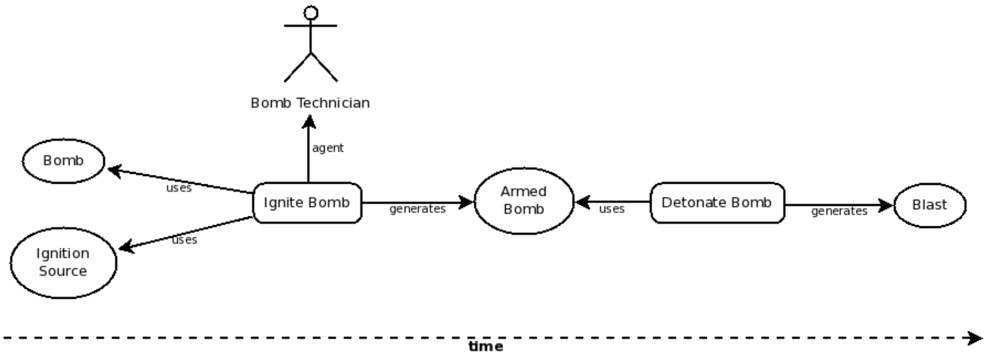


Figure 4.1: Depiction of a prospective description of a formal process (ellipses and rectangles denote artefacts and actions respectively), where a bomb will be ignited, and will subsequently explode.

Clearly, the role of an observer is more complex than that of an artefact. Artefacts are inert components of a system, whose state does not change without being subjected to an outside cause. In contrast, observers are agents of change within a system, whose actions affect the state of their environment. But then, consider the role of an exploding bomb; an inert component, until its fuse is lit (depicted in Figure 4.1). Furthermore, consider the role of a sensor; most definitely, sensors are inert components, until a stimulus is detected, and data is generated.

We argue that, in this context, the most suitable discriminant to distinguish between an observer and an artefact is the concept of "participation". An artefact is a passive participant, whose role (in the sequence of events) is completely determined by its interactions with other artefacts. In contrast, an observer is an active participant, whose actions are intentional, guided by sense perception, and have a measurable effect on the environment. Hence, we immediately infer that an observer is a specialisation of an artefact, i.e., an observer is an artefact, which may intentionally perform actions. Moreover, we infer that an exploding bomb is an artefact, and that a sensor is an observer.

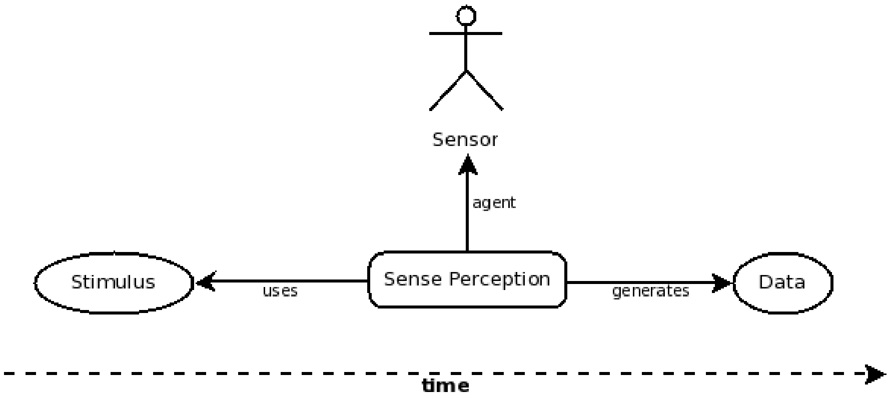


FIGURE 4.2: Depiction of a prospective description of a formal process (ellipses and rectangles denote artefacts and actions respectively), where a sensor will use sense perception in order to generate data.

Finally, we note that, a sensor does indeed modify the state of its environment. This is because, in this context, the "environment" is an instance of a mathematical model, whose "state" is an encoding of a possible configuration of a universe of information, which may itself include entities from both physical and digital reality (depicted in Figure 4.2). Thus, while the actions of a sensor do not affect the state of its physical environment, as new data is generated, a sensor does affect the state of its digital environment, i.e., the environment in which said data is represented and persisted.

4.1.5 Summary

In this section, we have presented a philosophical consideration of the nature of formal processes. Our work can be summarised by the following six principles.

PRINCIPLE 1: **Intention is actualised as action.**

To fully describe a formal process, it is necessary to invoke both a prospective and retrospective frame of reference, and to assert both our intentions and our actions, and to relate those assertions using the concept of actualisation.

PRINCIPLE 2: **Satisfaction is exceptional.**

By relating the retrospective and prospective descriptions of a formal process using the concept of actualisation, it is possible to distinguish between the cases of falling short of, satisfying, and exceeding one's expectations.

PRINCIPLE 3: **Deviation is the norm.**

A correct description of an incorrect entity is better than an incorrect description of a correct entity. Or said differently, we should always describe our actions, even if they are not "correct". Whether or not our original intentions have been satisfied may be determined retrospectively, given the context of a prospective description.

PRINCIPLE 4: **Repetition is really really good.**

No restriction is placed on the number of times that an individual action may be actualised. Hence, any action may be repeated, any number of times, where each repetition creates a distinct chain of events.

PRINCIPLE 5: **Active participation implies agency.**

An artefact is a passive participant, whose role (in the sequence of events) is completely determined by its interactions with other artefacts. In contrast, an observer is an active participant, whose actions are intentional, guided by sense perception, and have a measurable effect on the environment.

4.2 Ontology

In this section, we describe in detail the core component of this work: the Planning and Enactment (P&E) ontology. The ontology is serialised using the Web Ontology Language (OWL).

Prefix	Namespace URI	Description
pe	http://www.soton.ac.uk/~mib104/2012/10/26-pe-ns#	P&E terms

TABLE 4.1: Namespaces and prefixes used in Section 4.2.

4.2.1 Entities and Relationships

In Figure 4.3, we give a depiction of the core P&E entities and their inter-relationships. The entities are partitioned into two distinct sets, plan- and enactment-things, which correspond to prospective and retrospective frames of reference (discussed in Section 4.1.2). Relationships between entities in the same frame of reference are mirrored, i.e., for each relationship in the prospective frame of reference, there exists a corresponding relationship in the retrospective frame of reference.

The entities are defined as follows:

Plan – A prospective description of a formal process, which may be actualised in the future.

Plan-action – A prospective description of an action, which may be actualised in the future, as part of the actualisation of a plan.

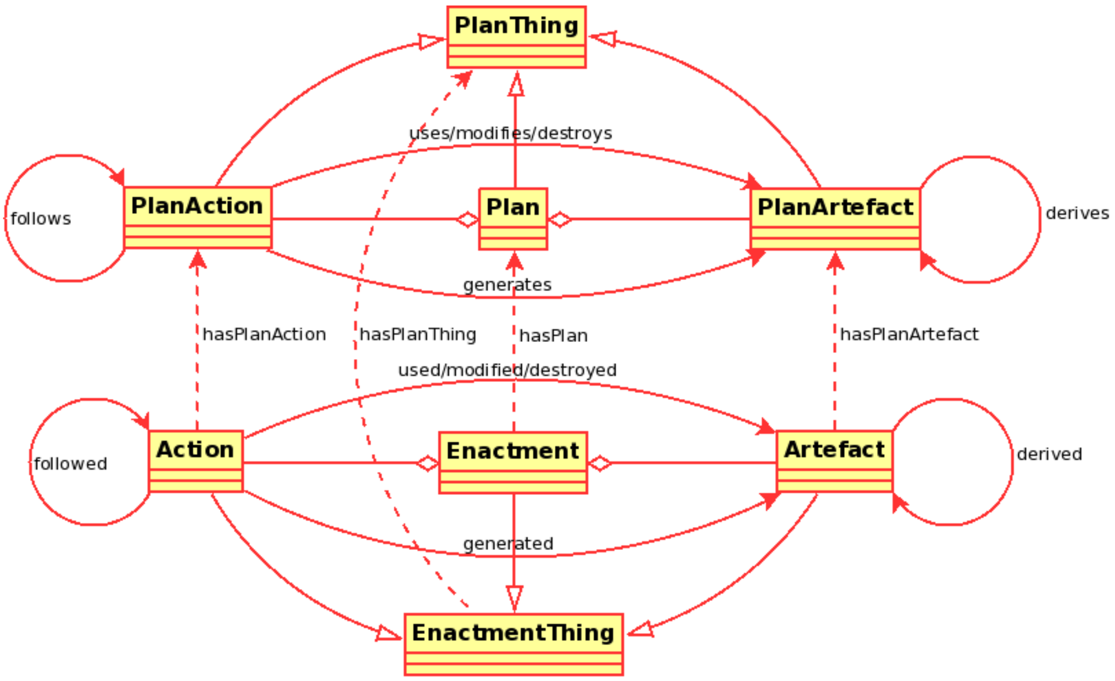


Figure 4.3: UML class diagram for the Planning and Enactment (P&E) ontology.

Plan-artefact – A prospective description of an artefact, which may be actualised in the future, as part of the actualisation of a plan.

Enactment – A retrospective description of the realisation of a formal process.

Action – A retrospective description of an action, which was realised as part of an enactment.

Artefact – A retrospective description of an artefact, which was realised as part of an enactment.

The actions and artefacts, which are contained in a plan or enactment, are related by predicates, which are grouped into six categories, based on their specialised semantics (listed in Table 4.2). Each category contains a total of four predicates (one verb for each frame of reference, and one inverse for each verb). The categories are summarised as follows:

Generation – Relates an action to an artefact that will be generated during the enactment of said action (prospective), or was generated during the enactment of said action (retrospective);

Utilisation – Relates an action to an artefact that will be used during the enactment of said action (prospective), or was used during the enactment of said action (retrospective);

Category	**Domain**	**Codomain**	**Prospective**	**Prospective Inverse**	**Retrospective**	**Retrospective Inverse**
Generation	Action	Artefact	generates (‡)	isGeneratedBy (†)	generated (‡)	wasGeneratedBy (†)
Utilisation	Action	Artefact	uses	isUsedBy	used	wasUsedBy
Modification	Action	Artefact	modifies	isModifiedBy	modified	wasModifiedBy
Destruction	Action	Artefact	destroys (‡)	isDestroyedBy (†)	destroyed (‡)	wasDestroyedBy (†)
Causation	Action	Action	follows (§)	isFollowedBy (§)	followed (§)	wasFollowedBy (§)
Lineage	Artefact	Artefact	derives (§)	isDerivedFrom (§)	derived (§)	wasDerivedFrom (§)

Table 4.2: Relationships between artefacts and actions in the Planning and Enactment (P&E) ontology. Additional properties of each relationship are given in parenthesis, where: (†) denotes being functional; (‡) denotes being inverse functional, and (§) denotes transitivity.

Modification – A specialisation of "utilisation", which relates an action to an artefact that will be modified during the enactment of said action (prospective), or was modified during the enactment of said action (retrospective);

Destruction – A specialisation of "modification", which relates an action to an artefact that will be destroyed during the enactment of said action (prospective), or was destroyed during the enactment of said action (retrospective);

Causation – Relates an action to another action, with the interpretation that the enactment of "Action #2" will follow that of "Action #1" (prospective), or that the enactment of "Action #2" followed that of "Action #1" (retrospective); and

Lineage – Relates an artefact to another artefact, with the interpretation that the characteristics of "Artefact #2" will derive from those of "Artefact #1" (prospective), or that the characteristics of "Artefact #2" are derived from those of "Artefact #1" (retrospective).

The predicates of the first four categories describe relationships between actions and artefacts, and are intended to be asserted *ab initio*. The predicates of the last two categories relate actions to other actions, and artefacts to other artefacts, and may be asserted either *ab initio* or *a priori*. In the case of the latter, we infer new assertions by evaluating inference rules.

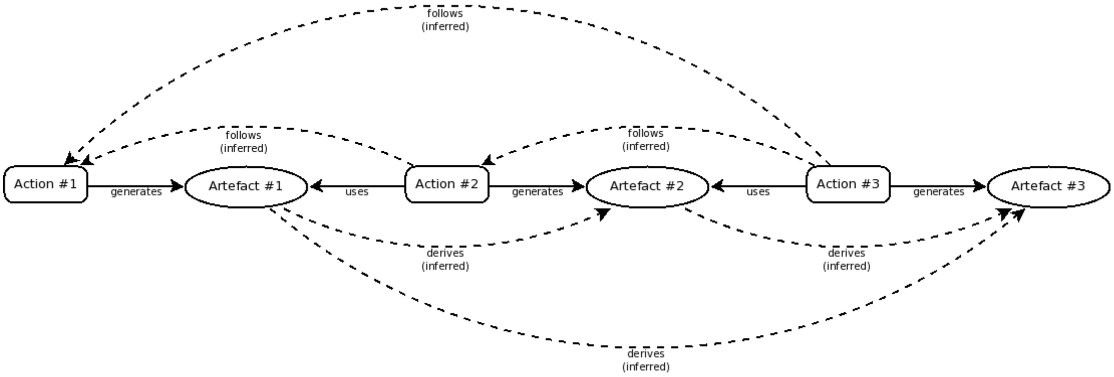


FIGURE 4.4: Depiction of asserted and inferred relationships between entities in an excerpt of a prospective description of a formal process (ellipses and rectangles denote artefacts and actions respectively).

A worked example, where inference results in the assertion of three new relationships, is given in Figure 4.4. The example depicts a prospective description of a formal process, which contains three actions and three artefacts, where the nth artefact is generated by the nth action, and used by the $n + 1$th action. Relationships that denote generation and utilisation of artefacts are asserted *ab initio*. Relationships that denote causation and lineage of actions and artefacts are asserted *a priori*.

4.2.1.1 Generation

Artefacts are generated during the enactment of an action. An artefact cannot be used before it has been generated. The predicates that represent the concept of "generation" are inverse-functional, i.e., during an enactment, an artefact is generated by exactly one action.

4.2.1.2 Utilisation, Modification and Destruction

Artefacts are used, modified and destroyed during the enactment of an action. An artefact cannot be used after it has been destroyed. The predicates that represent the concepts of "destruction" are inverse-functional, i.e., during an enactment, an artefact is destroyed by exactly one action.

4.2.1.3 Causation and Lineage

Predicates that represent the concepts of the "causation" and "lineage" of actions and artefacts are, by definition, transitive, i.e., if an artefact a is derived from another artefact b, which is itself derived from a third artefact c, then, under the transitive closure, we infer that a is derived from c. However, in our ontology, instead of denoting each concept by a single predicate, we actually define two distinct predicates.

For each concept, "causation" and "lineage", we define a predicate Φ, and a second, transitive predicate Φ^+, such that Φ implies Φ^+, and that Φ^+ is the transitive closure of Φ. There are two key advantages to this approach. First, the predicate Φ now denotes a *direct* relationship between its operands, which can be easily distinguished from the *indirect* relationship under transitive closure Φ^+. Second, because every assertion of Φ implies a corresponding assertion of Φ^+, the system suffers no information loss.

Causation (for prospective descriptions of actions) is expressed in English as follows:

> Given a prospective description P, for each pair of actions p and q, if p generates an artefact α that is used by q, then p is followed by q.

Formally, the predicate for causation (of prospective descriptions of actions) is expressed using first-order logic as follows:

$$\forall P \; \exists p \; \exists q \; \exists \alpha \left((p, q, \alpha \in P) \land \text{generates}\,(p, \alpha) \land \text{uses}\,(q, \alpha) \rightarrow \text{follows}\,(q, p) \right)$$

Similarly, lineage (for prospective descriptions of artefacts) is expressed in English as follows:

> Given a prospective description P, for each pair of artefacts α and β, if α is used by an action p that generates β, then α derives β.

Formally, the predicate for lineage (of prospective descriptions of artefacts) is expressed using first-order logic as follows:

$$\forall P \; \exists p \; \exists \alpha \; \exists \beta \left((p, \alpha, \beta \in P) \wedge \mathrm{uses}\,(p, \alpha) \wedge \mathrm{generates}\,(p, \beta) \rightarrow \mathrm{derives}\,(\alpha, \beta) \right)$$

For both concepts, we automatically obtain dual inference rules for retrospective descriptions:

$$\forall R \; \exists r \; \exists s \; \exists \alpha \left((r, s, \alpha \in R) \wedge \mathrm{generated}\,(r, \alpha) \wedge \mathrm{used}\,(s, \alpha) \rightarrow \mathrm{followed}\,(s, r) \right)$$

and

$$\forall R \; \exists r \; \exists \alpha \; \exists \beta \left((r, \alpha, \beta \in R) \wedge \mathrm{used}\,(r, \alpha) \wedge \mathrm{generated}\,(r, \beta) \rightarrow \mathrm{derived}\,(\alpha, \beta) \right)$$

In the context of W3C PROV, some authors have questioned[1] the validity of the inference of causation and lineage predicates, as described above. As an exemplar use case, they consider a formal process that consists of five artefacts a, b, c, d and e and exactly one action, which encapsulates the evaluation of two algebraic expressions $d = a + b$ and $e = b + c$. In their example, the authors note that the system infers six new predicates, which describe the lineage of each artefact. However, the authors also note that, given the naïve application of the rules, two of the inferences are invalid, i.e., it is not true that d is derived from c or that e is derived from a.

We argue that, in the above example, it is incorrect to blame the naïve application of the rules as, clearly, the cause of the incorrect inferences is the granularity of the description of the formal process itself, where, in this case, one action encapsulates more than one independent event. In this situation, it is our opinion that the correct approach is to encapsulate the evaluation of each algebraic expression as a separate action.

4.2.1.4 Actualisation and Satisfaction

In our ontology, the concept of "actualisation" (discussed in Section 4.1.3), which relates a retrospective description to a prospective description, is denoted (at least partially) by the `pe:hasPlanThing` predicate. We also define distinct predicates for each

[1] `http://www.w3.org/2001/sw/wiki/PROV-FAQ#Can_I_infer_Derivation_from_Usage_and_Generation.3F`

pair of prospective and retrospective classes: `pe:hasPlan`, `pe:hasPlanAction`, and `pe:hasPlanArtefact`.

It is important to note that the assertion of the `pe:hasPlanThing` predicate (or one of its specialisations) does not always imply that the domain (a retrospective description) is an actualisation of the codomain (a prospective description). For example, a retrospective description of an action is not truly an actualisation of a prospective description unless it has used and generated every artefact, and followed every preceding action.

This approach contrasts with that of W3C PROV (see Section 2.3.2.4), which specifies a “Plan” class to represent a description of a set of actions that were intended to be performed by an agent, and a “hadPlan” predicate to relate a retrospective description of a conceptual entity to its plan, but does not specify the nature of characteristics of plans.

Furthermore, it is interesting to compare the etymology of the “hadPlan” predicate in W3C PROV and the `pe:hasPlan` predicate of our ontology. In W3C PROV, the relationship between a retrospective description of a conceptual entity and its plan is an association, which indicates that [at some point in the life-cycle of said conceptual entity] a plan may [or may not] have been followed, i.e., that said retrospective description [may or may not have] had a plan. Whereas, in our ontology, the relationship corresponds to a true assertion of an actualisation event, i.e., that said retrospective description [definitely] has a plan.

This approach also contrasts with that of P-PLAN [93], an extension to W3C PROV, which aims to specify the representation and interpretation of plans. Like P-PLAN, our ontology provides a specific prospective class for each retrospective class. However, unlike P-PLAN, where the semantics of the association between retrospective and prospective entities is based on the concept of correspondence, in our ontology, the semantics of the association are based on actualisation.

4.2.2 Life Cycles

In this section, we describe the life-cycles for retrospective descriptions of artefacts and actions. In our ontology, the current state of an artefact or action can be determined by analysing the set of time-stamps, which are asserted by said artefact or action, where each time-stamp is denoted by the assertion of a specially-defined predicate. We define our own predicates, rather than reuse predicates from preexisting ontologies or vocabularies, for two key reasons. First, it is necessary to distinguish between assertions about artefacts and actions, and assertions about retrospective descriptions of those artefacts and actions. Hence, we define our own predicates, as they are automatically delineated by the P&E namespace. Second, although our predicates share the same names as those found in preexisting ontologies, they have a highly-specific semantics and interpretation. Thus, we define our own predicates, in order to enforce said semantics and interpretation.

4.2.2.1 Artefacts

In Figure 4.5, we give a depiction of a state machine that describes a retrospective description of an artefact. The state machine has three states, which correspond to the assertion of three time-stamps: "createdAt", "modifiedAt" and "destroyedAt". The state of an artefact is determined by analysing the asserted time-stamps, e.g., if the retrospective description of an artefact asserts the "destroyedAt" time-stamp, then the artefact is in the "Destroyed" state.

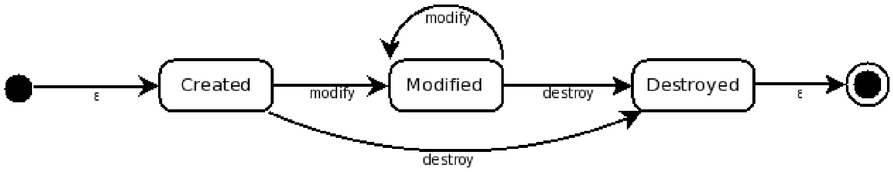


FIGURE 4.5: Depiction of the life-cycle of an artefact, as described by the Planning and Enactment (P&E) ontology (where ϵ denotes an epsilon transition).

For example, consider an artefact (of any type), which has existed in reality since a specific time t_0. At a specific time in the future $t_1 > t_0$, the artefact is observed and measured, and these measurements are realised as a retrospective description. The first time-stamp t_0 denotes the time at which the artefact itself was created. In contrast, the second time-stamp t_1 denotes the time at which the retrospective description of the artefact was created. Hence, the artefact is in the "Created" state.

At a specific time in the future $t_2 > t_1$, the artefact is observed and measured for a second time. Still further into the future, at time $t_3 > t_2$, the new measurements are realised as a second retrospective description, which is derived from the first. The third time-stamp t_2 denotes the time at which the artefact itself was modified. In contrast, the fourth time-stamp t_3 denotes the time at which the retrospective description of the modified artefact was created. Hence, the artefact is in the "Modified" state.

Finally, at a specific time in the future $t_4 > t_3$, the artefact is observed and measured for a third time. Still further into the future, at time $t_5 > t_4$, the new measurements are realised as a third retrospective description, which is derived from the second. The fifth time-stamp t_4 denotes the time at which the artefact itself was [considered to be] destroyed. In contrast, the sixth time-stamp t_5 denotes the time at which the retrospective description of the destroyed artefact was created. Hence, the artefact is in the "Destroyed" state.

In the above example, the effect of each transition is the creation of a new retrospective description, which is derived from a prior retrospective description. Each successive retrospective description is interpreted as a new revision, which either describes novel aspects of an artefact, or redefines preexisting aspects of an artefact. Hence, the current state of an artefact at a given time t_n can only be determined by analysis of the combination of all prior retrospective descriptions of said artefact.

4.2.2.2 Actions

In Figure 4.6, we give a depiction of a state machine that describes a retrospective description of an action. The state machine has seven states, which correspond to the assertion of six time-stamps: "pendingAt", "readyToStartAt", "startedAt", "readyToFinishAt", "finishedAt", and "cancelledAt". The state of an action is determined by analysing the asserted time-stamps, e.g., if the retrospective description of an action asserts the "cancelledAt" time-stamp, then the action is in the "Cancelled" state. For the remainder of this section, the agent that manages the enactment is referred to as "the system".

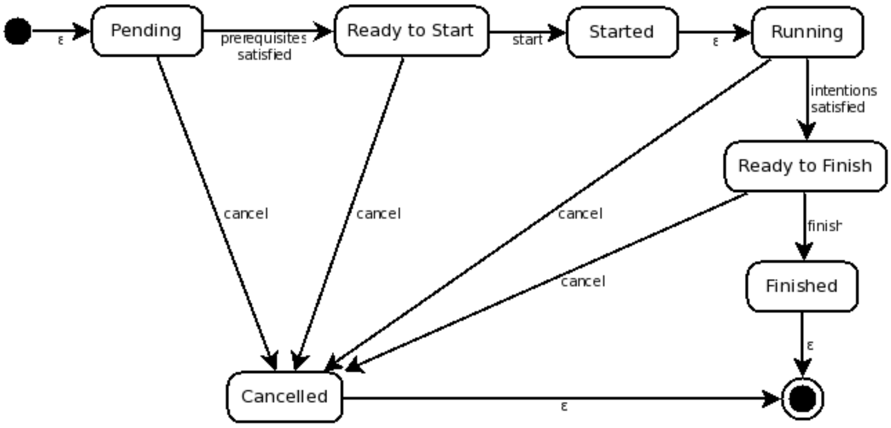


FIGURE 4.6: Depiction of the life-cycle of an action, as described by the Planning and Enactment (P&E) ontology (where ϵ denotes an epsilon transition).

For example, consider an action (of any type). At a specific time in the future t_0, a retrospective description of the action is realised, i.e., an information resource is constructed as a place-holder, and said information resource is allocated an identifier. The first time-stamp t_0 denotes the time at which the action was pending enactment. Hence, the action is in the "Pending" state.

At a specific time in the future $t_1 > t_0$, the system decides (for some reason) that either the dependencies for the enactment of the action have been satisfied, or that the enactment of the action should be cancelled. If the former is true, then the second time-stamp t_1 denotes the time at which the enactment of the action was ready to start, and the action is in the "Ready to Start" state. Otherwise, if the latter is true, then the second time-stamp t_1 denotes the time at which the enactment of the action was cancelled, and the action is in the "Cancelled" state.

At a specific time in the future $t_2 > t_1$, the system decides (for some reason) that either the enactment of the action should be started, or that the enactment of the action should be cancelled. If the former is true, then the third time-stamp t_2 denotes the time at which the enactment of the action was started, and the action is in the “Started” state. Otherwise, if the latter is true, then the third time-stamp t_2 denotes the time at which the enactment of the action was cancelled, and the action is in the “Cancelled” state.

Given the presence of an epsilon transition[2], an action that is in the “Started” state automatically moves into the “Running” state.

At a specific time in the future $t_3 > t_2$, the system decides (for some reason) that either the original intentions for the enactment have been satisfied, or that the enactment of the action should be cancelled. If the former is true, then the fourth time-stamp t_3 denotes the time at which the enactment of the action was ready to finish, and the action is in the “Ready to Finish” state. Otherwise, if the latter is true, then the fourth time-stamp t_3 denotes the time at which the enactment of the action was cancelled, and the action is in the “Cancelled” state.

At a specific time in the future $t_4 > t_3$, the system decides (for some reason) that either the enactment of the action has finished, or that the enactment of the action should be cancelled. If the former is true, then the fifth time-stamp t_4 denotes the time at which the enactment of the action was finished, and the action is in the “Finished” state. Otherwise, if the latter is true, then the fifth time-stamp t_4 denotes the time at which the enactment of the action was cancelled, and the action is in the “Cancelled” state.

In the above example, the effect of each transition is the assertion of a new time-stamp as part of the retrospective description of the action, i.e., there is no need to create a new retrospective description after each transition.

4.2.2.3 Enactments

In our ontology, enactments are interpreted as combinations of actions, i.e., in a sense, enactments are macro-scale actions. Thus, the state machine that describes the life-cycle of an enactment is identical to that of an action.

4.2.3 Assumptions

In this subsection, we list the assumptions that were made during the development of the ontology. We discuss the implications of each assumption, and provide a resolution strategy for any issues that are discovered, where each resolution strategy is an extension module or “plug-in” for the core ontology.

[2] In a state machine, an epsilon transition is one that may be optionally followed.

4.2.3.1 The Enactment Environment (Space)

In our ontology, the actualisation of a formal process occurs within a space, which is referred to as the "enactment environment" (discussed in Section 4.1.3). However, in the core ontology, we do not specify a conceptual entity to represent the concept of a space, nor do we define a predicate to relate things to locations.

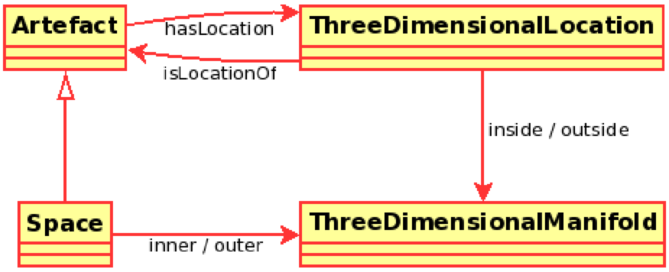


FIGURE 4.7: UML class diagram for an extension to the Planning and Enactment (P&E) ontology, which defines the concepts of the enactment environment (a space) and location.

In Figure 4.7, we give the UML class diagram for an extension to the P&E ontology, which defines (for the retrospective frame of reference only) the concepts of spaces and locations within spaces.

The extension models a space as an artefact, which is delineated from other spaces by a boundary (that may or may not have thickness). Each space is defined by either one or two three-dimensional manifolds; one for the inner surface of the boundary of the space, and another for the outer surface of the boundary of the space. Thus, if the inner and outer manifolds are defined to be identical, then the boundary of the space has zero thickness, otherwise, the boundary of the space has non-zero, positive thickness.

The concept of an artefact's location within a space is modelled by a special-purpose "location" entity, whose role is to encapsulate the vector of the artefact's co-ordinates in three-space. Thus, given a location and a manifold, it is trivial to determine if said location is either inside or outside said manifold. However, it is important to note that, for the purposes of this work, we assume a model that uses gauge fixing, i.e., the location of each artefact is specified according to a global co-ordinates system, with a fixed point of origin.

This approach contrasts with that of W3C PROV (see Section 2.3.2.4), which specifies a "Location" class, but does not restrict its characteristics or representation, e.g., in W3C PROV, the "Location" class is disjoint to all other classes, and instances may be constructed for any identifiable place.

4.2.3.2 Agents are Artefacts

In our ontology, we have argued that agents should be interpreted as specialised artefacts; specifically, an agent is an artefact that can intentionally perform actions (discussed in Section 4.1.4).

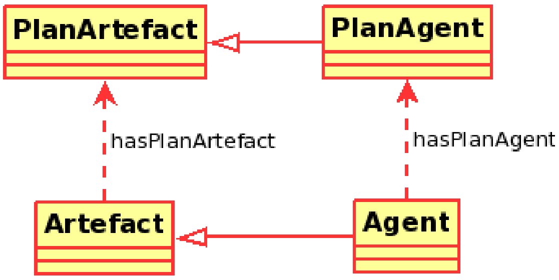


Figure 4.8: UML class diagram for an extension to the Planning and Enactment (P&E) ontology, which defines the concept of an agent.

In Figure 4.8, we give the UML class diagram for an extension to the P&E ontology, which defines (for both the prospective and retrospective frames of reference) the concept of an agent. Interestingly, a consequence of the proposed extension is that the set of relationships (between artefacts and actions) are immediately reusable. For example, as all agents are artefacts, an agent may be specified as a prerequisite for the actualisation of an action; and, moreover, as a prerequisite, an agent may be inferred as the anti-derivative for any artefacts that were generated during the actualisation of an action (discussed in Section 4.2.1.3).

Our approach contrasts with those of OPM and W3C PROV (see Section 2.3.2.4), which both specify that the agent and artefact classes are disjoint.

4.2.3.3 Annotations are Artefacts

In the previous section, we described how, as an extension to the core ontology, agents may be interpreted as specialised artefacts. In this section, we argue that annotations should also be interpreted as specialised artefacts, which are necessarily disjoint to agents.

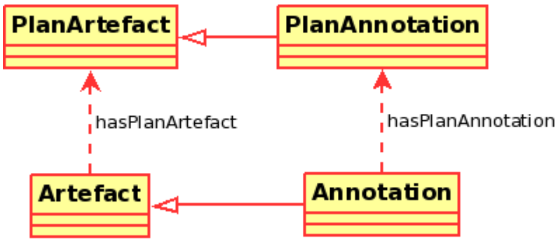


Figure 4.9: UML class diagram for an extension to the Planning and Enactment (P&E) ontology, which defines the concept of an annotation.

In Figure 4.9, we give the UML class diagram for an extension to the P&E ontology, which defines (for both the prospective and retrospective frames of reference) the concept of an annotation. A key implication of this approach is that, using our ontology, it is possible to distinguish between annotations that were generated intentionally or unintentionally, i.e., annotations that were generated either with or without the context of a prospective description.

A natural consequence of our approach is that, as they are always generated within the context of a prospective description, e.g., the source code, all annotations that are generated by software systems must be intentional. In contrast, annotations that are generated by a human being may or may not be intentional, depending on the context that is provided by the prospective description.

4.2.3.4 Reification of Retrospective Relationships

In our ontology, inter-entity relationships are asserted using binary predicates, where each assertion is a tuple of a label, a domain and a codomain, which is interpreted according to the denotational semantics of the aforementioned label. As each tuple contains only three elements, no additional information is provided by an assertion.

For prospective descriptions, where all entities are are assumed to be endurants, which are demarcated from other entities by a container (the plan), this restriction raises no issues. However, for retrospective descriptions, where all entities are assumed to be perdurants, whose interpretations may have one or more temporal qualities, this restriction raises a subtle issue: at what time was each assertion asserted?

For example, consider an assertion of the “generated” relationship, which relates a retrospective description of an action (the domain) to a retrospective description of an artefact (the codomain), and is interpreted to mean that, at some point in time during the actualisation of the domain, the codomain was actualised. However, given the current definition of the ontology, it is not possible to assert the specific time at which a “generation” event occurred. Instead, we must assume that one of the following alternatives is true:

- That the codomain was actualised at the start of the actualisation of the domain;
- That the codomain was actualised at the end of the actualisation of the domain; or
- That the codomain was actualised between the start and end of the actualisation of the domain.

Given the context, the first and second alternatives are clearly wrong. Firstly, it is not possible for the codomain to be actualised at the same instant as the start of the actualisation of the domain, as a finite, but non-zero, period of time must pass between the cause (the actualisation of the domain) and the effect (the actualisation of the codomain). Moreover, it is not possible for the codomain to be actualised at the same instant as the end of the actualisation of the codomain, as, obviously, the actualisation has ended, and, therefore, no more events can occur. Thus, we must assume that the third alternative is true.

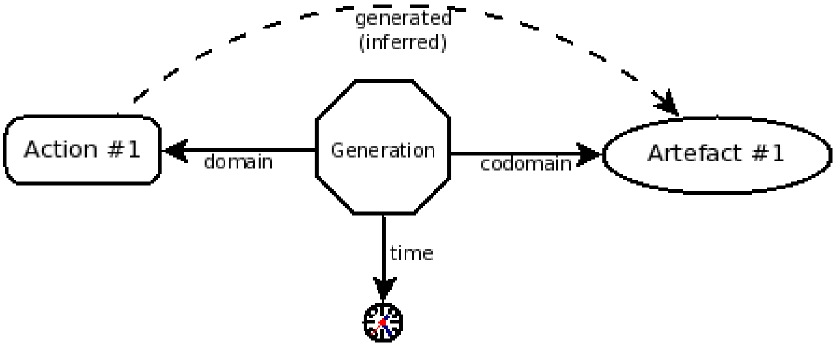


Figure 4.10: Depiction of asserted and inferred relationships between entities in an excerpt of a retrospective description of a formal process (ellipses, rectangles and octagons represent artefacts, actions and reifications respectively).

Clearly, this situation is not satisfactory. The most pertinent issue is that, given the current definition of the ontology, the truth of the assumption is not testable, i.e., it is not possible to determine if the codomain was actually actualised during the actualisation of the domain. A more practical approach would be to reify the "generated" relationship, and model it as a distinct entity (depicted in Figure 4.10). There are three key advantages to this approach. First, as an entity, the reification may relate any number of other entities. Second, the reification may assert additional information, such as time-stamps. Third, the original relationship (a binary predicate) may be recovered by inference.

4.2.4 Summary

In this section, we have presented the entities of the P&E ontology, their inter-relationships, and life-cycles. We have described the semantics and given an interpretation for each relationship, and provided a worked example for the life-cycle of each entity. Furthermore, we have listed our assumptions for the design and design rationale of the ontology, and given suggestions for future work.

Finally, in Figure 4.11, we give a depiction of the graph of the entities and relationships of the P&E ontology, which was rendered using a force-directed layout algorithm. We note that, the application of a force-directed algorithm results in a figure with a high degree of visual symmetry, reflecting the symmetrical definitions of prospective and retrospective entities and relationships in the ontology.

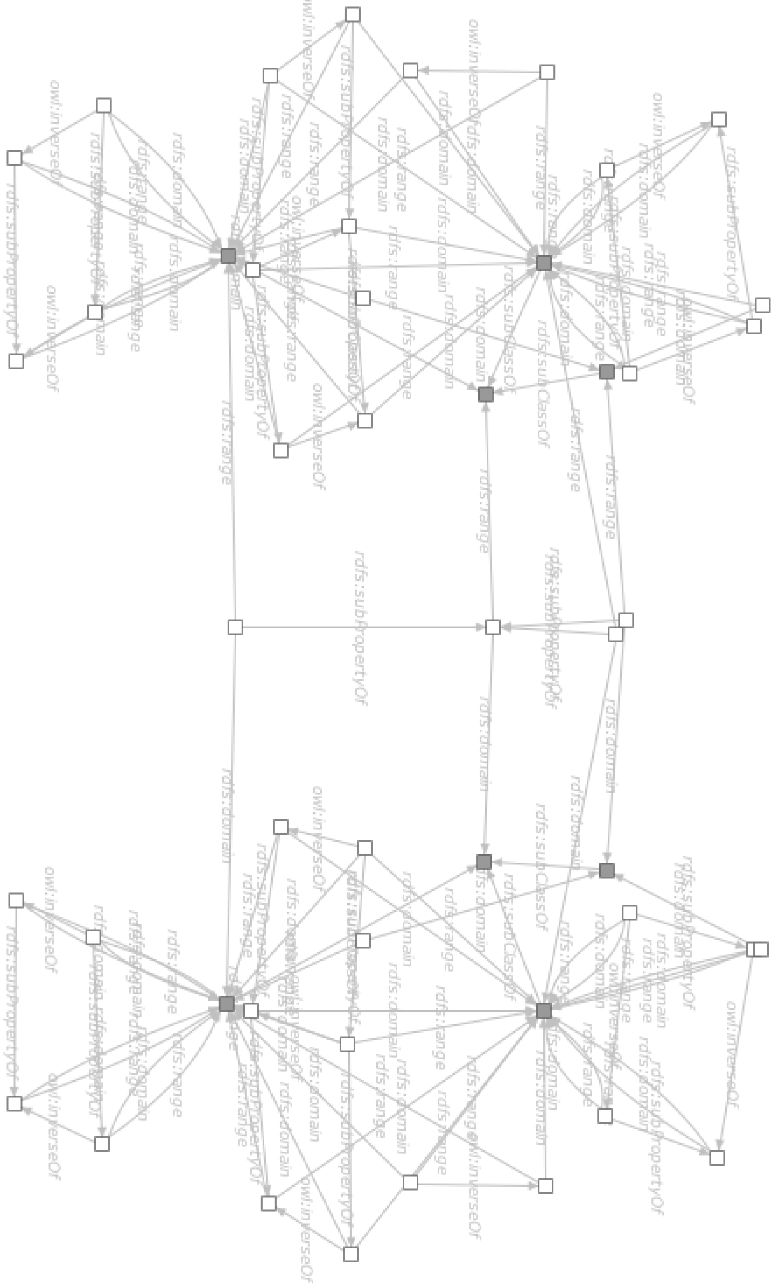


Figure 4.11: Depiction of the Planning and Enactment (P&E) ontology. Nodes representing classes and predicates are coloured grey and white respectively.

4.3 Integration with eCrystals Repository for Crystal Structures

In this section, we describe the integration of the Planning and Enactment (P&E) ontology with the eCrystals[3] repository for crystal structures.

Please note that the work [114] that is reported in this section was completed as part of the oreChem[4] project, which ran from October 2009 to October 2011. Following its completion, the outputs of the oreChem project were repurposed, for use as the basis for the P&E ontology. Hence, for consistency, with respect to the rest of this chapter, this section uses P&E terminology.

eCrystals is a repository for crystal structures [115], which are generated by the Southampton Chemical Crystallography Group (SCCD), and the EPSRC UK National Crystallography Service (NCS)[5]. The repository is designed according to open-access principles, i.e., following a mandatory embargo period, all records are publicly accessible. This is intended to facilitate independent verification and validation of the determined crystal structures by interested third-parties.

Each record in the repository is an aggregation of the fundamental and derived data that is generated during the enactment of a crystal structure determination workflow; a formal process, which is enacted partly *in vivo* and *in silico*. The workflow begins when a new sample of an unknown chemical substance is received. Using an X-ray source and diffractometer, technicians collect raw data about the crystal structure of the sample. The raw data is processed and refined using a variety of specialised software applications, until, eventually, the crystal structure is determined. Finally, a new record is uploaded to the repository.

Since its deployment, nearly 800 records have been uploaded to the repository. However, as the details of the crystal structure determination workflow are not disseminated in a machine-processable format, the retrospective provenance information for data files that are aggregated by each record cannot be determined, e.g., it is not possible to determine the software application that generated a particular data file. Hence, the primary goal of this work is to construct a machine-processable representation of the crystal structure determination workflow using the P&E ontology, and to enhance the software system that underlies the repository, such that new records are disseminated with additional contextual information, and in a semantically-rich form. Furthermore, the secondary goal of this work is to investigate whether or not our techniques may also be applied to pre-existing records, i.e., the use of the machine-processable representation of the crystal

[3] http://ecrystals.chem.soton.ac.uk/

[4] http://research.microsoft.com/en-us/projects/orechem/

[5] http://www.ncs.ac.uk/

structure determination workflow as the basis for inference of new information about pre-existing records.

As we have alluded throughout, a key motivation for this work is to facilitate the timely detection of negative and/or fraudulent results. In the domain of crystallography, where large portions of research are automated by the use of software applications, achieving this functionality is particularly challenging, as, at any one time, the complete lineage for each result must be considered, in order for any anomalies to be detected, e.g., in the context of the crystal structure determination, the raw data from the detector and the determined crystal structure may be in agreement with each other, but the intermediate results may have been falsified [85, 116].

In Figure 4.12, we present a depiction of the prospective description of the crystal structure determination workflow (referred to as the "plan"), which was developed in collaboration with experts from both the SCCD and NCS, and published via the repository. Accordingly, the plan may be downloaded from the following publicly-accessible location:

`http://ecrystals.chem.soton.ac.uk/plan.rdf`

In Figure 4.13, we present a depiction of the retrospective description of the partial enactment of the crystal structure determination workflow for a single record. The retrospective description describes the sequence of software applications that were executed, along with the data files that were used as inputs, and generated as outputs. Each entity in the retrospective description is related to a corresponding entity in the plan, i.e., the retrospective description of the execution of each software application is related to a descriptor for said software application, and the retrospective description of the utilisation and/or generation of each data file is related to a descriptor for said data file.

This approach has the key advantage that consumers may query the system for all actualisations of a specific data file, rather than for all data files of a specific format. The implication of this approach is that consumers may generate complex and powerful queries, e.g., "find all actualisations of the specified data file descriptor" or "find all data files that were generated from an actualisation of the specified data file descriptor." Moreover, such queries may themselves be combined, into yet more powerful queries, e.g., given two data file descriptors a and b, consumers may query for all actualisations whose lineage includes a and whose progeny includes b.

In Figure 4.14, we give an example of such a query (encoded using SPARQL), which could be used as the basis for the implementation of an automatic crystal structure verification service. The query, which is grounded by the eCrystals plan, returns a set of quads, where each quad includes a reference to an enactment, along with references to a raw, intermediate, and reported data file that was used and/or generated during said enactment. Within the context of the plan, the raw data file is an actualisation of the "HKL" descriptor (commonly referred to as the reflection data file, where h, k and ℓ are the Miller indices for the lattice planes of the crystal structure); and, the reported data file is an actualisation of the "CIF" descriptor (where CIF stands for Crystallographic Information File). The intermediate data files are exactly as their name suggests, i.e., intermediate, and, hence, can only be described in terms their relationships to the raw and reported data files.

In Figure 4.15, we present a depiction of the retrospective description of the partial enactment of the crystal structure determination workflow for a single record, where the lineage of each data file has been automatically inferred (discussed in Section 4.2.1.3). To facilitate navigation and information discovery, in our extension to the eCrystals repository, said "data lineage graphs" are presented as interactive image-maps, where the depiction of each vertex is hyperlinked to its corresponding data file.

In this section, we have described the integration of the P&E ontology with the eCrystals repository for crystal structures. The integration takes the form of a software extension for the repository that encapsulates a prospective description of the crystal structure determination workflow, which was designed in collaboration with domain experts. The prospective description is used by the software extension to automatically associate retrospective provenance information with each record in the repository; both pre-existing and new deposits. Finally, the retrospective provenance information is used in order to construct graphical depictions, which are hyperlinked to data files, and shared with end-users.

4.4 Plan for the Enactment of Plans

In this section, we present a prospective description of a formal process (a plan), whose actualisation constitutes the actualisation of another plan, i.e., a plan that describes the enactment of other plans.

The goal of this work is three-fold. First, to identify a methodology for actualisation of prospective descriptions of formal processes, and to codify said methodology using the entities and relationships that are defined by the P&E ontology. Second, to inform the design of the P&E ontology, by identifying aspects of the methodology that are not describable, given the current definition of the P&E ontology. Finally, to inform the design and implementation of software systems that relate to the realisation of the methodology.

Before continuing, it is important to note that, while heuristics for the synthesis and analysis of methodologies have been explored in detail by other authors, most notably by

Pólya [117], the computable representation of such methodologies has not. Accordingly, the contribution of this section is to demonstrate how such methodologies may be defined and used within a machine-processable framework (in this case, the P&E ontology).

In Figure 4.16, we give a depiction of a methodology for the actualisation of a prospective description of a formal process. The methodology is a composition of three sub-methodologies:

- A methodology for entering and leaving spaces;
- A methodology for starting and finishing the actualisation of a plan; and
- A methodology for starting and finishing the actualisation of a plan-action.

In English, the methodology is as follows:

1. Start the enactment.
2. ≪*decision*≫ Enter a space; or, do nothing, and go to (4).
3. Repeat the following:
   (a) ≪*decision*≫ Select a plan; or, leave the space, and go to (2).
   (b) Start the enactment of the selected plan.
   (c) Repeat the following:
      i. Assess the state of the space.
      ii. Aggregate the plan-actions whose prerequisites have been satisfied.
      iii. ≪*decision*≫ Actualise a plan-action; or, do nothing, and go to (3d).
      iv. Audit the actualisation of the plan-action.
   (d) Finish the enactment of the selected plan.
4. Finish the enactment.

We make the following observations:

- A record of the enactment is created regardless of whether or not a space is eventually entered, or if a plan is eventually actualised;
- The decision to enter (or leave) a space is modelled explicitly, i.e., the system that implements the methodology must be informed of the presence (or absence) of each artefact in the space;
- The methodology does not restrict the number of plans that may be enacted;

- The methodology does not restrict which plan may be selected, i.e., the decision to actualise a plan is the responsibility of the participant, and not of the system that implements the methodology;
- The actualisation of the selected plan starts immediately, i.e., a record of the enactment of the selected plan is created regardless of whether or not any plan-actions are eventually actualised;
- The sub-methodology for the enactment of a plan-stage is stateless, i.e., the state of the system is that of the space;
- The methodology does not restrict which plan-actions may be selected, i.e., the decision to actualise a plan-action is the responsibility of the participant, and not of the system that implements the methodology. Instead, the methodology stipulates that the prerequisites for each plan-action are satisfied; and
- The methodology does not specify its own termination conditions, i.e., the decision to finish the actualisation of a plan is the responsibility of the participant, and not of the system that implements the methodology.

We note that, as specified by the methodology, the system must record which spaces are entered (and exited), which plans and plan-actions are selected for actualisation, and the effects of each actualisation. The key advantage of this approach is that, as the system is able to distinguish between intention and action, expectation and outcome, the assertions that describe the actualisation of each entity may be interpreted with full context. For example, consider the pathological case, where a participant enters a space with the intention to actualise a given plan, but, for whatever reason, he subsequently learns that the actualisation is not possible. In this situation, a record of the actualisation of the plan is created, but, said record is empty, i.e., it contains no descriptions of actualisations of plan-actions. Hence, the system may distinguish between the distinct cases of "not doing anything" and "not being able to do anything".

We have described how the methodology foregoes restriction, in order to facilitate the actualisation of any sequence of actions. Moreover, we have noted how the definition of the methodology does not, under any circumstances, attempt to prescribe which actions are "correct" or "appropriate". Furthermore, instead of prescribing a set of termination conditions, the methodology simply allows participants to "do nothing", and terminate any enactment (that has been started, but is neither finished nor cancelled) at will. Given this framework, participants are free to impose their own set of termination conditions, e.g., participants may decide to stop when the selected prospective description has been satisfied, or, perhaps, when they are simply too tired to continue working!

Finally, we have described in detail how the methodology allows one to deviate from one's original intentions, e.g., by performing additional, or unspecified, plan-actions.

However, we now proceed to consider the concept of “improvisation” – the performance of actions without prior preparation. In this context, the retrospective description of formal processes, an improvisation is simply a retrospective description that is an actualisation of the null prospective description.

4.5 Summary

In this chapter, we have presented an ontology for the exposition of both the prospective and retrospective provenance of formal processes, which are enacted both *in silico* and *in vivo*.

We have presented a philosophical consideration of the nature of formal processes, and encapsulated our findings as a set of principles. We have concluded that, in order to fully describe a formal process, it is necessary to invoke both the prospective and retrospective frames of reference. Furthermore, we have shown that if, within the context of an ontology, it is consistent to describe negative results, then it is also consistent to describe all results, i.e., to describe scenarios where the original intentions of the observer were either failed, satisfied, or exceeded.

We have presented a machine-processable ontology for the description of formal processes, which is defined by an entity-relationship model, and a set of a production rules: the Planning and Enactment (P&E) ontology. During the development of the ontology, we found that, in order to describe the location of event, it is necessary to incorporate the concept of a three-dimensional manifold into our model. Furthermore, we found that, in order to formally describe the time at which an event occurs, it is necessary to abandon the concept of point-like events, and, instead, to model the occurrence of each event as a non-zero duration.

In collaboration with the Southampton Chemical Crystallography Group (SCCD) and the EPSRC UK National Crystallography Service (NCS), we have enhanced the eCrystals repository for crystal structures by providing retrospective provenance descriptions for every record (including new submissions). The goal of the work was to demonstrate that a prospective description of the eCrystals workflow could be constructed, and that said description could be used by an automated software system in order to generate a retrospective description for each record. Our approach was successful, and the enhancements have been in production use since May 2010. Furthermore, the retrospective provenance descriptions have been repurposed, in order to generate graphical depictions of the data lineage graph for each record, where each node in the graph is hyperlinked to the corresponding data file.

Using the ontology, we have constructed a prospective description of a formal process (a plan), whose actualisation constitutes the actualisation of another plan, i.e., a plan that describes the enactment of another plan. The construction of the “meta-plan” is a demonstration of the possibility of implementing a generic software system for the planning and enactment of formal processes.

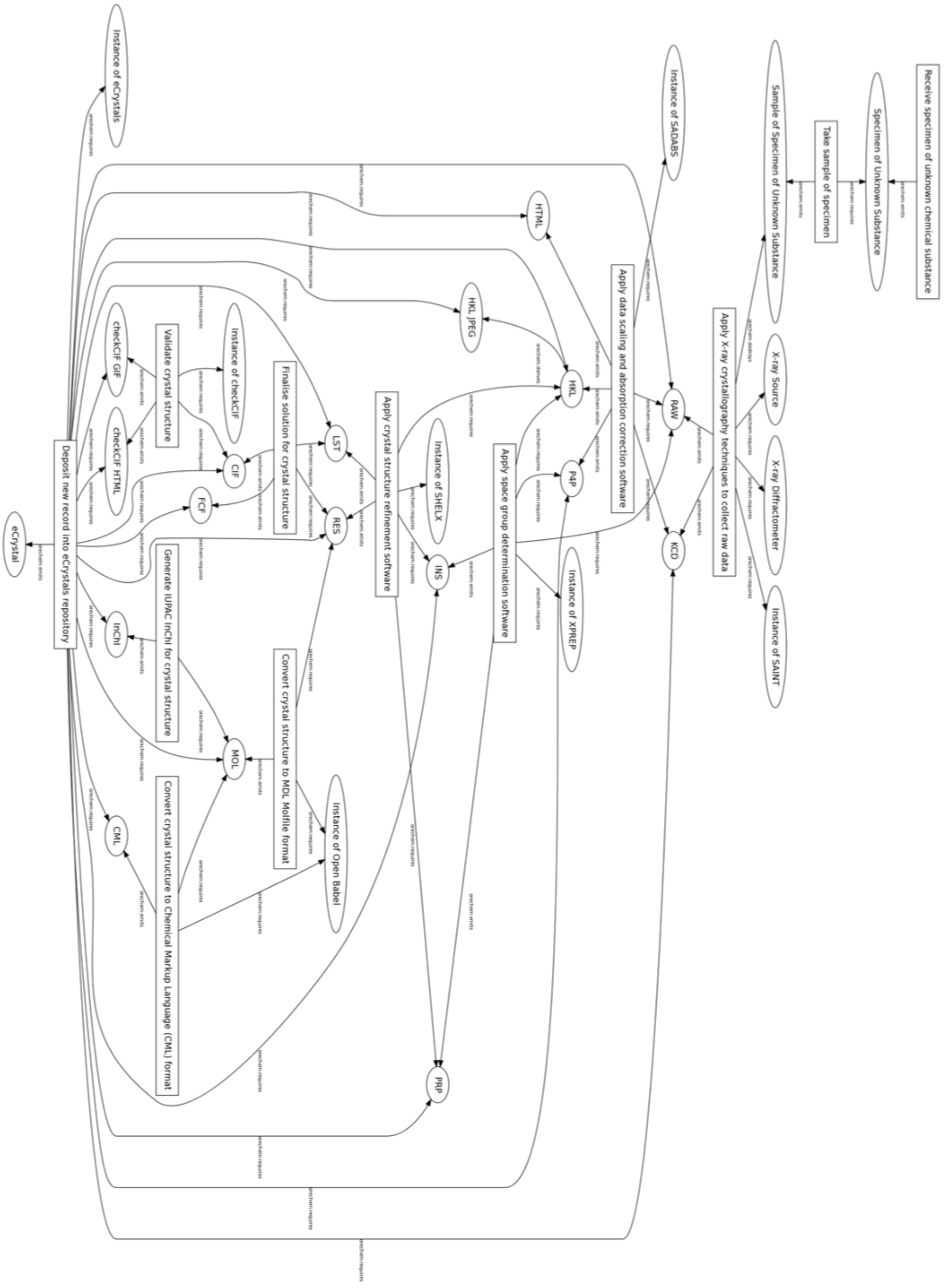

Receive specimen of unknown chemical substance
Specimen of Unknown Substance
Take sample of specimen
Sample of Specimen of Unknown Substance
Apply X-ray crystallography techniques to collect raw data
X-ray Source
X-ray Diffractometer
Instance of SAINT
Instance of SADABS
RAW
KCD
Apply data scaling and absorption correction software
HKL
P4P
HTML
HKL JPEG
Apply space group determination software
Instance of XPREP
Apply crystal structure refinement software
Instance of SHELX
INS
PRP
LST
RES
Finalise solution for crystal structure
CIF
FCF
Instance of checkCIF
Validate crystal structure
checkCIF GIF
checkCIF HTML
Generate IUPAC InChI for crystal structure
InChI
MOL
Convert crystal structure to MDL Molfile format
Instance of Open Babel
Convert crystal structure to Chemical Markup Language (CML) format
CML
Deposit new record into eCrystals repository
Instance of eCrystals
eCrystal

FIGURE 4.12: Depiction of the prospective description of the eCrystals crystal structure determination workflow, described in terms of the oreChem Core Ontology (the precursor to the Planning and Enactment (P&E) ontology). Rectangles and ellipses correspond to software applications and data files respectively. Available at: http://ecrystals.chem.soton.ac.uk/plan.rdf

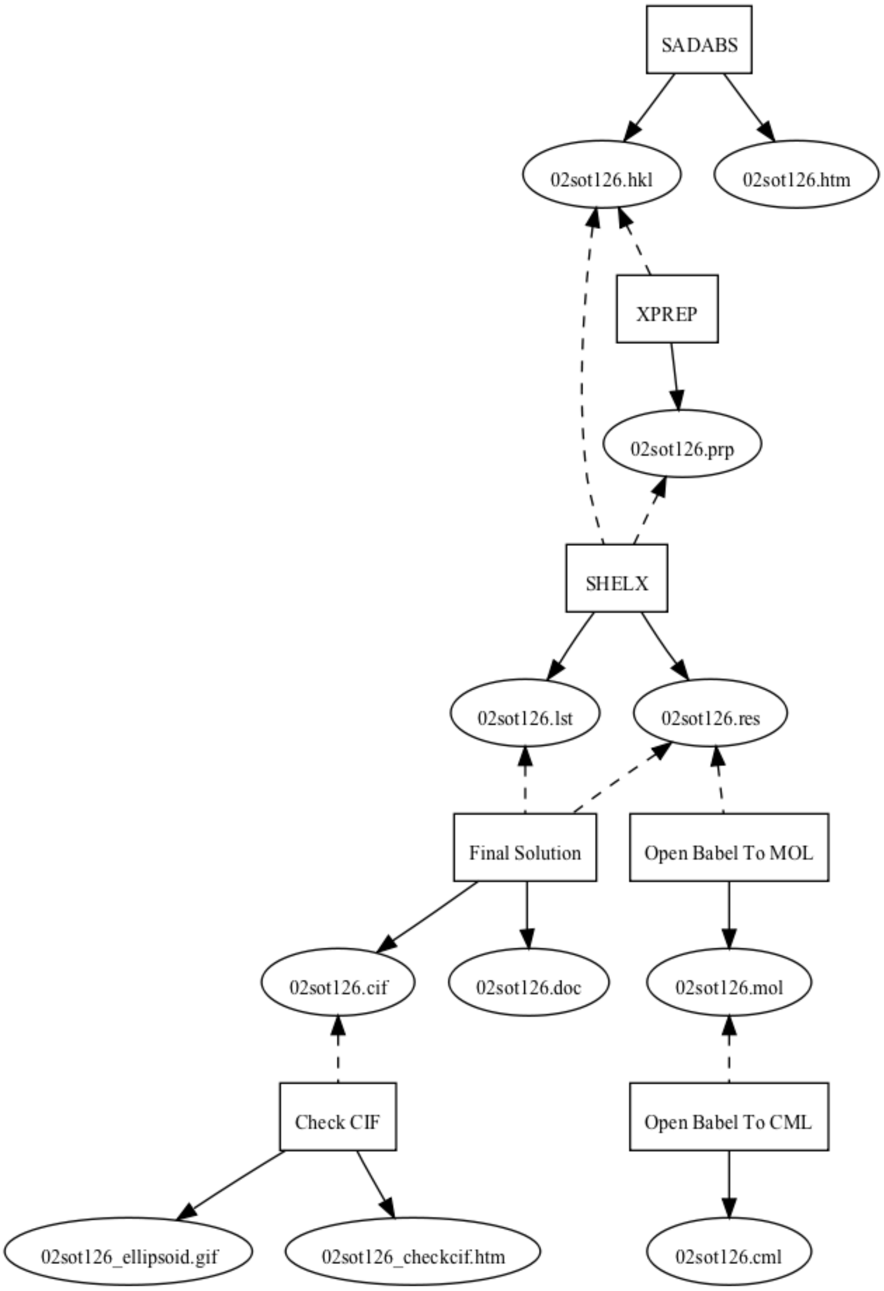


Figure 4.13: Depiction of the retrospective description of the partial enactment of the eCrystals crystal structure determination workflow, for record #29, where rectangles and ellipses correspond to software applications and data files respectively, and solid and dashed edges correspond to assertions of the `orechem:emitted` and `orechem:used` predicates. Available at: http://ecrystals.chem.soton.ac.uk/cgi/export/29/ORE_Chem/ecrystals-eprint-29.xml

```
PREFIX orechem: <http://www.openarchives.org/2010/05/24-orechem-ns#>
PREFIX plan: <http://ecrystals.chem.soton.ac.uk/plan.rdf>
SELECT ?run ?raw_data_file ?intermediate_data_file ?reported_data_file
FROM <http://ecrystals.chem.soton.ac.uk/cgi/export/29/ORE_Chem/ecrystals-eprint-29.xml>
WHERE {
  ?run
    orechem:hasPlan plan:Ecrystals ;
    orechem:containsObject ?raw_data_file ;
    orechem:containsObject ?intermediate_data_file ;
    orechem:containsObject ?reported_data_file .

  ?raw_data_file
    orechem:hasPlanObject plan:HKL .

  ?intermediate_data_file
    orechem:derivedFrom ?raw_data_file .

  ?reported_data_file
    orechem:hasPlanObject plan:CIF ;
    orechem:derivedFrom ?intermediate_data_file .
}
```

FIGURE 4.14: SPARQL query that returns a set of quads, where each quad includes a reference to a retrospective description of the enactment of a formal process, along with references to the raw, intermediate, and reported data files that were used and/or generated during said enactment.

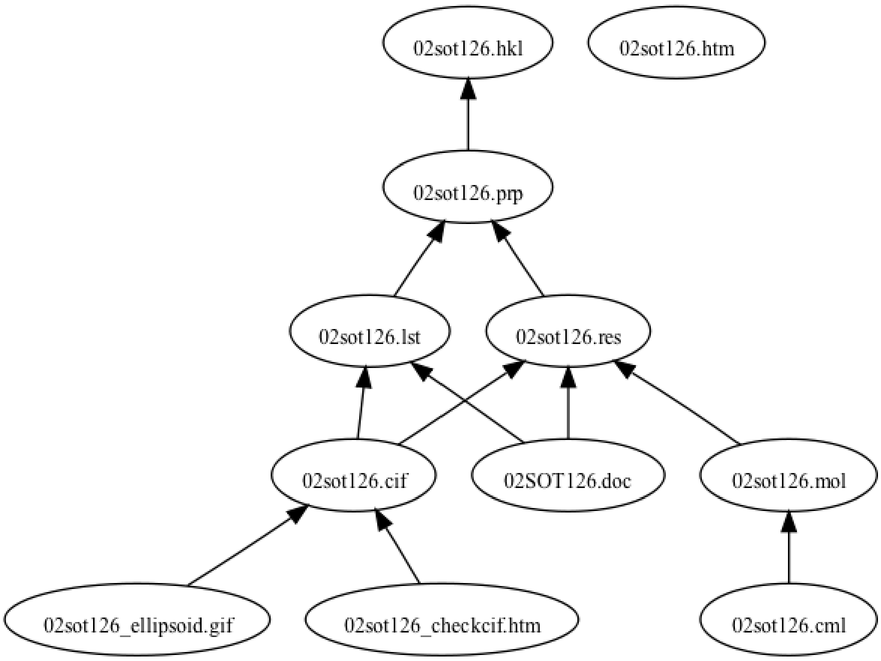


FIGURE 4.15: Depiction of the retrospective description of the partial enactment of the eCrystals crystal structure determination workflow, for record #29, where each ellipse corresponds to a data file, and edges correspond to assertions of the `orechem:derivedFrom` predicate. Available at: http://ecrystals.chem.soton.ac.uk/cgi/export/29/ORE_Chem_Provenance/ecrystals-eprint-29.png

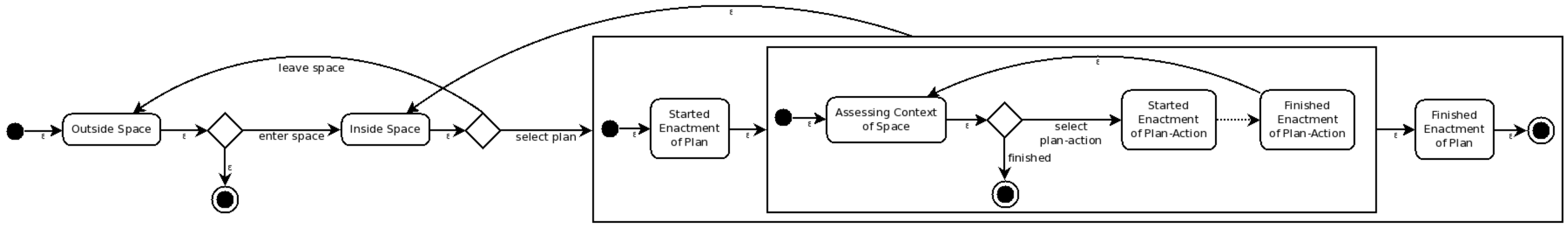

ε
Outside Space
ε
enter space
ε
Inside Space
ε
select plan
leave space
ε
ε
Started Enactment of Plan
ε
ε
Assessing Context of Space
ε
select plan-action
finished
Started Enactment of Plan-Action
Finished Enactment of Plan-Action
ε
ε
Finished Enactment of Plan
ε

Figure 4.16: Depiction of the flow diagram for a plan that describes the realisation of another plan (where ϵ denotes an epsilon-transition).

Chapter 5

Conclusions

"T'ain't what you do (it's the way that you do it)"
– James Young and Sy Oliver (1939)

To conduct research is to actualise the scientific method – the cyclic process of planning and enacting scientific experiments, acquiring data, synthesising information, and organising knowledge. For many years, the primary record of research has been the paper-based laboratory notebook. However, as more researchers are deciding to augment their workflows with the use of software systems, the dominant position of the paper-based laboratory notebook appears to be under threat. In the context of this transition, this thesis has explored how knowledge representation techniques and technologies may be used in order to augment the scientific method, and hence, may be used in order to support research.

In Section 2.1, we discussed the concept of a knowledge representation technology, which we characterised as a mechanism for encoding descriptions of conceptual entities as symbolic messages, such that they may be communicated with automata. However, we argued that, as an automaton neither "understands" the meaning of messages, nor "infers" the content of new messages from old messages, its capabilities, to recognise and/or manipulate the symbols of a message, are determined by the characteristics of the knowledge representation technologies that are used.

We discussed a prominent use of knowledge representation technologies: the Semantic Web – a movement, which argues for the inclusion of machine-processable data in Web documents. Two of the core technologies of the Semantic Web are the Resource Description Framework (RDF; a family of specifications, which collectively define a methodology for the modelling and representation of information resources as structured data) and the Web Ontology Language (OWL; an extension of RDF, which is used to describe the semantics of ontologies and other entity-relationship models). We also discussed the concept of Linked Data – a set of techniques and best practices for the dissemination of structured data on the Web using RDF, OWL and related technologies. Finally, in Section 2.1.3, we discussed well-known RDF resources and OWL ontologies that could be used to support research, including: controlled vocabularies for the assertion of bibliographic metadata, personal information and the aggregation of heterogeneous information resources; and, ontologies for knowledge organisation and dataset description.

In Section 2.2, we discussed the concept of a laboratory notebook, which we characterised as a class of content management system (CMS). We argued that the value proposition for the use of laboratory notebooks by researchers is twofold. First, that the laboratory notebook provides the researcher with a mechanism for performing cognitive delegation, i.e., the laboratory notebook assists with the enactment of the researcher's workflow. Second, by ensuring that it is accessible, manipulable, meaningful, understandable, etc., that the laboratory notebook provisions for the realisation of the ephemeral value of its content. Given these capabilities, researchers may share and reuse the content of each other's laboratory notebooks, i.e., perform data integration.

We considered three aspects of laboratory notebooks: their capabilities; the characteristics of their content; and, the mediums, which are used for their implementation. We argued that, as the overwhelming majority of the capabilities of a CMS are generic, the specialisation of a CMS is a response to the specialisation of its content (and not *vice versa*). Hence, we argued that, to establish a *lingua franca* between researchers and software systems, it is of vital importance that the characteristics of the content (its structure, semantics, life-cycle, etc.) be formalised.

Next, we considered the characteristics of the content of a laboratory notebook, and noted that, while the structure and semantics of the content are derived from the ontology and nomenclature of the domain of discourse, the context of the content is derived from the circumstances that surrounded its generation:

Plan – The specification of the process, which will be enacted, by which the content will be generated, i.e., prospective provenance information.

Enactment – The description of the process, which was enacted, by which the content was generated, i.e., retrospective provenance information.

Hyperlinks – The assertion of the relationships between the prospective and retrospective provenance information, i.e., the correspondence (or non-correspondence) between the intentions and actions of the participants, e.g., actualisation, instruction, interpretation and inspiration.

Hence, we argued that, as it is independent of any specific domain of discourse, and therefore, is generic, the capture and curation of provenance information is a capability of the CMS. Thus, we identified an opportunity to provide a formalisation of the specification and actualisation of such processes (hereinafter referred to as "formal processes").

Finally, we considered the two dominant mediums for the implementation of laboratory notebooks: paper and software; and noted that, while the they share many capabilities, they have one fundamental difference: the dis- or collocation of the physical and logical information of their content. For paper-based laboratory notebooks, we found that the physical information of the paper is disjoint to the logical information that is encoded on the surface(s) of the paper, and therefore, that the content of a paper-based laboratory notebook is always consistent, regardless of its correctness, i.e., that anything that is capable of being encoded on the surface of a sheet of paper may be encoded, e.g., that we may write $2 + 2 = 5$, with no negative effects (except for the author's embarrassment).

In contrast, for an electronic laboratory notebook (ELN; the digitisation of the laboratory notebook concept), we found that the physical and logical information of the content are identical, and hence, are collocated. Thus, an ELN (or CMS in general) must be able to distinguish between content that is correct and content that is encodable, otherwise, it risks responding to that which is incorrect as if it were inconsistent. However, we noted that, unlike encodability, the correctness of content is wholly determined by its provenance information, and hence, if sufficient provenance information is captured and curated, then the correctness of content can be determined retrospectively.

In Section 2.3, we highlighted two relevant applications of knowledge representation techniques and technologies: computational workflows; and, the formalisation of the capture and curation of provenance information. We noted that, as they are specifications of formal processes, whose purpose is to orchestrate the manipulation of data by other software systems, the capabilities of computational workflows, like those of the CMS, are determined by the characteristics of the data. Furthermore, we noted that, the semantics of a computational workflow are such that, when enacted, any deviation from the specification, such as a failure of an associated software system or the generation of data that does not conform to the requirements of an interface, means that the enactment as a whole is considered as incorrect. Of course, we noted that, while this behaviour is entirely suitable for computations, where the zealous adherence to a specification is a functional requirement, it is not always suitable for other types of formal process, where deviation and/or improvisation may be required, e.g., scientific experiments, recipes and artistic performances. Indeed, during our analysis, we noted that the majority of existing systems for the formalisation of provenance information were implemented to support the use of computational workflows, and hence, that there is an opportunity to provide a formalisation of the "intent-centred" provenance information of other types of formal process.

In Chapter 3, we presented three new datasets: a controlled vocabulary of quantities, units and symbols that are used in physical chemistry, derived from the subject index IUPAC Green Book, using RDF; a controlled vocabulary for the classification and labelling of potentially hazardous chemical substances, derived from the Globally Harmonized System of Classification and Labelling of Chemicals (GHS), using RDF and OWL; and, a Linked Data interface for the RSC ChemSpider online chemical database. To demonstrate the use of the GHS dataset, we presented a Web-based software application, which automates the task of generating health and safety assessment forms. We found that, generally, machine-processable representations of controlled vocabularies and other nomenclature are beneficial to researchers, as they are a source of domain-specific identifiers, which can be used to reference concepts unambiguously, and hence, facilitate data integration.

To complement the development of the software application, we conducted a cost-benefit analysis for its deployment within an organisation that employs individuals to perform scientific experiments. We found that, while the organisation is likely to see benefits from the deployment, such as a reduction in operating costs, the individual is likely to be against the deployment, due to the belief that it will invert the direction of accountability within the organisation, and hence, increase their personal level of legal responsibility. Interestingly, we also found that an important factor is the presence of an association with an authoritative, well-known or trusted brand.

Throughout Chapter 3, we argued for the development of machine-processable datasets and software systems that can be used by laboratory-based researchers to both support their research and to provision for the ephemeral value of their scholarly works. However, we also identified the many risks that arise when such products are used naïvely, where the truth (or falsity) of the assertions, and the validity of the generated artefacts, are blindly accepted, without any proof or explanation. Hence, we proposed that the most sustainable mechanism for the mitigation of these risks is the formal exposition of provenance information.

In Chapter 4, we presented an OWL ontology for the exposition of the provenance of formal processes – the Planning and Enactment (P&E) ontology. Our motivation for this work was to provide a detailed consideration of the nature and characteristics of formal processes; and, to formalise the specification and actualisation of domain-specific formal processes. To demonstrate the robustness of our approach, we developed a machine-processable representation of the eCrystals repository for crystal structures, where a partial description of the retrospective provenance of each record is automatically inferred from a prospective description of the formal process for generating a new record.

In Section 4.4, we identified that a consequence of our approach is the specification of a prospective description of a formal process whose actualisation constitutes the actualisation of another formal process, i.e., a “meta-plan” or plan for the enactment of another plan. Moreover, we outlined how, if such a formal process were encapsulated by a software system, then it would be possible to implement a provenance-aware space, where the retrospective provenance information of events that occur within may be (semi-)automatically recorded. Furthermore, we have shown that, given the generality of our approach, it would be possible to specialise both the software system and the provenance-aware space for specific domains of discourse, i.e., to implement provenance-aware laboratories, kitchens, performance spaces, construction sites, operating theatres, retail environments, etc.

In conclusion, to provision for the realisation of the ephemeral value of the content of a laboratory notebook (or CMS in general), the circumstances that surrounded the generation of said content must be captured and curated. If context is exposed, then the correctness of content can be determined retrospectively, and the causes and consequences of incorrect content can be identified, e.g., deviation from a specified plan, improvisation without specifying a plan, or actualisation of an erroneous plan. Together with the machine-processable representation of domain-specific nomenclature, the exposition of provenance information enables data integration, facilitates the dissemination and reuse of content, and hence, affords new capabilities to the researcher. For knowledge representation techniques and technologies to support research, the process of conducting the scientific method must be formalised, and thus, made computable. To achieve this, the developers of knowledge representation techniques and technologies must accept that there is a difference between theory and practice, and that it is not what you do, but the way that you do it.

5.1 Research Outcomes

Chapter 1 described eight novel research contributions in this thesis. The outcomes of these contributions are as follows:

1. The design of a controlled vocabulary for quantities, units and symbols that are used in physical chemistry, based on the IUPAC Green Book, has enabled the unambiguous communication of domain-specific information resources amongst physical chemists. It has also facilitated an assessment of the value proposition for the introduction of domain-specific nomenclature in generic software systems.

2. The design of a controlled vocabulary for classification and labelling of potentially hazardous chemical substances, based on the Globally Harmonized System of Classification and Labelling of Chemicals (GHS), has enabled the integration of data sources from disparate scientific software systems.

3. Augmenting the RSC ChemSpider online chemical database to support Linked Data has allowed RSC ChemSpider to take advantage of data that has been exported by other Linked Data projects. Allowing users to export and query over this data has given them the opportunity to integrate RSC ChemSpider with other data sources.

4. Implementation of a fully-automated health and safety assessment form generator has facilitated the exploration of the development and deployment of a disruptive technology, enabling the investigation of the value propositions and legal implications for such deployments.

5. Consideration of the nature and characteristics of formal processes has enabled the definition of a consistent logical framework, in which specific implementations have been aligned. It has also allowed us to explore the tacit assumptions that are made by specific implementations.

6. The design of an ontology for the exposition of both the prospective and retrospective provenance of formal processes facilitates the communication of both the intentions and actions of the researcher, allowing other researchers to interpret results with their complete and unambiguous context.

7. The specification of a formal process for the enactment of another formal process (that is described in terms of the ontology) is a demonstration of the completeness of our approach. Moreover, such a “meta-plan” could be used as the basis for the implementation of a generic software system for the planning and enactment of formal processes, which in turn could be used as a software-based assistant when inside of a provenance-aware space.

8. Augmenting the eCrystals repository for crystal structures to support Linked Data, specifically, the retrospective provenance information of database records, has both facilitated and enhanced communication with end-users. Allowing users to export and query over this data has given them the opportunity to investigate the process by which crystal structures were determined, and hence, has given them a greater degree of trust in the data.

5.2 Further Work

As this chapter has made clear, there are several areas in which the work undertaken by this thesis can be extended:

1. The techniques developed in Section 3.1 could be reapplied for the machine-processable representation of other domain-specific nomenclature, e.g., the subject indices of the remaining IUPAC “coloured books”.

2. The dataset presented in Section 3.2 could be extended to describe the classification and labelling of other chemical substances. Moreover, the dataset could be enhanced to describe chemical reactions.

3. The dataset presented in Section 3.3 could be enhanced to provide machine-processable descriptions of other aspects the RSC ChemSpider database, e.g., hyperlinks to associated searches, spectra, patents, publications and pharmacological data; and, vendor-specific chemical identifiers.

4. The software application presented in Section 3.2 could be enhanced to generate health and safety assessment forms using user-customisable templates.

5. The ontology presented in Section 4.2 could be harmonised with existing an provenance formalisation, e.g., developed as a module for PROV (see Section 2.3.2.4).

6. The "meta-plan" presented in Section 4.4 could be actualised as a software system.

Appendix A

Contents of CD-ROM

The following can be found on the CD-ROM that accompanies this thesis:

- Data and source code for the Globally Harmonized System of Classification and Labelling of Chemicals (GHS) dataset (see Section 3.2);
- Data and source code for the IUPAC Green Book dataset (see Section 3.1);
- Source code for the Planning and Enactment (P&E) ontology (see Section 4.2).

The directory tree for the CD-ROM is as follows:

```
/
├── chemistry
│   ├── ghs
│   │   ├── data
│   │   └── src
│   └── iupac
│       ├── data
│       └── src
└── pe
```

Bibliography

[1] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):28–37, May 2001.

[2] S. Bratt. Toward a web of data and programs. In *Proceedings of the 22nd IEEE Conference on Mass Storage Systems and Technologies*, pages 124–128. IEEE Computer Society, 2005.

[3] J.J. Carroll and G. Klyne. Resource Description Framework (RDF): Concepts and abstract syntax. W3C Recommendation, W3C, February 2004. Available at: http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.

[4] P. Hayes. RDF semantics. W3C Recommendation, W3C, February 2004. Available at: http://www.w3.org/TR/2004/REC-rdf-mt-20040210/.

[5] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. DBpedia – A crystallization point for the Web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154–165, 2009.

[6] T. Berners-Lee. Notation 3 logic: An RDF language for the Semantic Web. W3C Draft, W3C, August 2005. Available at: http://www.w3.org/DesignIssues/Notation3.html.

[7] D. Beckett and J. Grant. RDF test cases. W3C Recommendation, W3C, February 2004. Available at: http://www.w3.org/TR/2004/REC-rdf-testcases-20040210/.

[8] D. Beckett and T. Berners-Lee. Turtle – Terse RDF triple language. W3C Team Submission, W3C, March 2011. Available at: http://www.w3.org/TeamSubmission/2011/SUBM-turtle-20110328/.

[9] D. Beckett. RDF/XML syntax specification (revised). W3C Recommendation, W3C, February 2004. Available at: http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/.

[10] M. Lanthaler and C. Gütl. On using JSON-LD to create evolvable RESTful services. In *Proceedings of the 3rd International Workshop on RESTful Design*, pages 25–32. ACM, 2012.

[11] R.V. Guha and D. Brickley. RDF vocabulary description language 1.0: RDF Schema. W3C Recommendation, W3C, February 2004. Available at: http://www.w3.org/TR/2004/REC-rdf-schema-20040210/.

[12] P. Hayes, P.F. Patel-Schneider, and I. Horrocks. OWL web ontology language semantics and abstract syntax. W3C Recommendation, W3C, February 2004. Available at: http://www.w3.org/TR/2004/REC-owl-semantics-20040210/.

[13] C. Welty, D.L. McGuinness, and M.K. Smith. OWL web ontology language guide. W3C Recommendation, W3C, February 2004. Available at: http://www.w3.org/TR/2004/REC-owl-guide-20040210/.

[14] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. SWRL: A semantic web rule language combining OWL and RuleML. W3C Member Submission, W3C, May 2004. Available at: http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/.

[15] H. Boley, S. Tabet, and G. Wagner. Design rationale of RuleML: A markup language for semantic web rules. In *Proceedings of the International Semantic Web Working Symposium 2001*, pages 381–402, 2001.

[16] E. Prud'hommeaux and A. Seaborne. SPARQL query language for RDF. W3C Recommendation, W3C, January 2008. Available at: http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/.

[17] C. Bizer, T. Heath, and T. Berners-Lee. Linked data – the story so far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.

[18] T. Berners-Lee. Linked data – design issues. Available at: http://www.w3.org/DesignIssues/LinkedData, 2006.

[19] C. Bizer, A. Jentzsch, and R. Cyganiak. State of the LOD cloud. Available at: http://www4.wiwiss.fu-berlin.de/lodcloud/state, 2011.

[20] A. Powell, M. Nilsson, A. Naeve, P. Johnston, and T. Baker. DCMI abstract model. DCMI Recommendation, Dublin Core Metadata Initiative, 2007. Available at: http://dublincore.org/documents/2007/06/04/abstract-model/.

[21] DCMI Usage Board. DCMI metadata terms. DCMI Recommendation, Dublin Core Metadata Initiative, 2012. Available at: http://dublincore.org/documents/2012/06/14/dcmi-terms/.

[22] DCMI Usage Board. Dublin core metadata element set, version 1.1. DCMI Recommendation, Dublin Core Metadata Initiative, 2012. Available at: http://dublincore.org/documents/2012/06/14/dces/.

[23] D. Brickley and L. Miller. FOAF vocabulary specification 0.98. Namespace Document, FOAF Project, August 2010. Available at: http://xmlns.com/foaf/spec/20100809.html.

[24] F. Abel, N. Henze, E. Herder, and D. Krause. Interweaving public user profiles on the web. In P. Bra, A. Kobsa, and D. Chin, editors, *User Modeling, Adaptation, and Personalization*, volume 6075 of *Lecture Notes in Computer Science*, pages 16–27. Springer Berlin Heidelberg, 2010.

[25] U. Bojars, J.G. Breslin, D. Berrueta, D. Brickley, S. Decker, S. Fernández, C. Görn, A. Harth, T. Heath, K. Idehen, et al. SIOC core ontology specification. W3C Member Submission, W3C, June 2007. Available at: http://www.w3.org/Submission/2007/SUBM-sioc-spec-20070612/.

[26] C. Lagoze, H. Van de Sompel, P. Johnston, M. Nelson, R. Sanderson, and S. Warner. Primer. ORE User Guide, Open Archives Initiative, October 2008. Available at: http://www.openarchives.org/ore/1.0/primer.

[27] C. Lagoze, H. Van de Sompel, P. Johnston, M. Nelson, R. Sanderson, and S. Warner. Resource map implementation in Atom. ORE User Guide, Open Archives Initiative, October 2008. Available at: http://www.openarchives.org/ore/1.0/atom.

[28] C. Lagoze, H. Van de Sompel, P. Johnston, M. Nelson, R. Sanderson, and S. Warner. Resource map implementation in RDF/XML. ORE User Guide, Open Archives Initiative, October 2008. Available at: http://www.openarchives.org/ore/1.0/rdfxml.

[29] C. Lagoze, H. Van de Sompel, P. Johnston, M. Nelson, R. Sanderson, and S. Warner. Resource map implementation in RDFa. ORE User Guide, Open Archives Initiative, October 2008. Available at: http://www.openarchives.org/ore/1.0/rdfa.

[30] C. Lagoze, H. Van de Sompel, P. Johnston, M. Nelson, R. Sanderson, and S. Warner. Abstract data model. ORE Specification, Open Archives Initiative, October 2008. Available at: http://www.openarchives.org/ore/1.0/datamodel.

[31] S. Bechhofer, I. Buchan, D. De Roure, P. Missier, J. Ainsworth, J. Bhagat, P. Couch, D. Cruickshank, M. Delderfield, I. Dunlop, et al. Why linked data is not enough for scientists. *Future Generation Computer Systems*, 2011.

[32] E. Summers and A. Isaac. SKOS simple knowledge organization system primer. W3C Note, W3C, August 2009. Available at: http://www.w3.org/TR/2009/NOTE-skos-primer-20090818/.

[33] S. Bechhofer and A. Miles. SKOS simple knowledge organization system reference. W3C Recommendation, W3C, August 2009. Available at: http://www.w3.org/TR/2009/REC-skos-reference-20090818/.

[34] K. Alexander, R. Cyganiak, M. Hausenblas, and J. Zhao. Describing linked datasets with the VoID vocabulary. W3C Recommendation, W3C, March 2011. Available at: http://www.w3.org/TR/2011/NOTE-void-20110303/.

[35] R.D. Tweney. Faraday's notebooks: the active organization of creative science. *Physics Education*, 26(5):301, 1991.

[36] M. H. Elliott. The state of the ELN market. In *Laboratory Informatics Guide*, volume 91 of *Scientific Computing World*, pages 53–54. Europa Science, 2007. Available at: http://www.scientific-computing.com/features/feature.php?feature_id=50.

[37] J.D. Myers, T.C. Allison, S. Bittner, B. Didier, M. Frenklach, W.H. Green Jr, Y.L. Ho, J. Hewson, W. Koegler, L. Lansing, et al. A collaborative informatics infrastructure for multi-scale science. In *Proceedings of the 2nd IEEE International Workshop on Challenges of Large Applications in Distributed Environments*, pages 24–33. IEEE Computer Society, 2004.

[38] T. Talbott, M. Peterson, J. Schwidder, and J.D. Myers. Adapting the electronic laboratory notebook for the semantic era. In *Proceedings of the IEEE International Symposium on Collaborative Technologies and Systems*, pages 136–143. IEEE Computer Society, 2005.

[39] J.D. Myers, E.S. Mendoza, and B. Hoopes. A collaborative electronic laboratory notebook. In *Proceedings of the International Conference on Internet and Multimedia Systems and Applications*, pages 334–338. ACTA Press, 2001.

[40] C. Pancerella, J. Hewson, W. Koegler, D. Leahy, M. Lee, L. Rahn, C. Yang, J.D. Myers, B. Didier, R. McCoy, et al. Metadata in the collaboratory for multi-scale chemical science. In *Proceedings of the 3rd International Conference on Dublin Core and Metadata Applications*, DCMI '03, pages 13:1–13:9. Dublin Core Metadata Initiative, 2003.

[41] A. Tabard, W.E. Mackay, and E. Eastmond. From individual to collaborative: the evolution of Prism, a hybrid laboratory notebook. In *Proceedings of the 13th ACM Conference on Computer Supported Cooperative Work*, CSCW '08, pages 569–578, New York, NY, USA, 2008. ACM.

[42] J.G. Frey and M.B. Hursthouse. From e-science to publication@source. Presented at: National Policies on Open Access (OA) Provision for University Research Output: An International Meeting. University of Southampton, Southampton, UK, February 2004.

[43] m.c. schraefel, G. Hughes, H.R. Mills, G. Smith, and J.G. Frey. Making tea: Iterative design through analogy. In *Proceedings of the 5th Conference on Designing Interactive Systems*, pages 49–58, 2004.

[44] m.c. schraefel, G. Hughes, H.R. Mills, G. Smith, T. Payne, and J.G. Frey. Breaking the book: Translating the chemistry lab book into a pervasive computing lab environment. In *Proceedings of the 22nd Conference on Human Factors in Computing Systems*. ACM Press, 2004.

[45] K.R. Taylor, J.W. Essex, J.G. Frey, H.R. Mills, G. Hughes, and E.J. Zaluska. The semantic grid and chemistry: Experiences with Comb*e*Chem. *Web Semantics: Science, Services and Agents on the World Wide Web*, 4(2):84–101, 2006.

[46] Health and Safety Executive (HSE). *Control of substances hazardous to health (COSHH)*. HSE Books, 5th edition, 2005.

[47] A. McNaught. The IUPAC international chemical identifier. *Chemistry International*, 2006.

[48] N. Goddard, R. Macneil, and J. Ritchie. eCAT: Online electronic lab notebook for scientific research. *Automated Experimentation*, 1(1):4, 2009.

[49] V. Venkatasubramanian. DROWNING IN DATA: Informatics and modeling challenges in a data-rich networked world. *AIChE Journal*, 55(1):2–8, 2009.

[50] P. McKenzie, S. Kiang, J. Tom, A.E. Rubin, and M. Futran. Can pharmaceutical process development become high tech? *AIChE Journal*, 52(12):3990–3994, 2006.

[51] L.X. Yu. Pharmaceutical quality by design: Product and process development, understanding, and control. *Pharmaceutical Research*, 25(4):781–791, 2008.

[52] A.J.G. Hey and A.E. Trefethen. The data deluge: An e-science perspective. In F. Berman, G.C. Fox, and A.J.G. Hey, editors, *Grid Computing - Making the Global Infrastructure a Reality*, pages 809–824. Wiley and Sons, 2003.

[53] D. De Roure and J.G. Frey. Three perspectives on collaborative knowledge acquisition in e-science. In *Proceedings of the Workshop on Semantic Web for Collaborative Knowledge Acquisition*, 2007.

[54] B.P. Allen. Linked data standards and infrastructure for scientific publishing. In *Proceedings of the W3C Workshop on Linked Enterprise Data Patterns*, 2011.

[55] P. Lord, A. Macdonald, L. Lyon, and D. Giaretta. From data deluge to data curation. In *Proceedings of the 3rd UK e-Science All Hands Meeting*, pages 371–375, 2004.

[56] J. Zhao, R.D. Stevens, C.J. Wroe, M. Greenwood, and C.A. Goble. The origin and history of in silico experiments. In *Proceedings of the 3rd UK e-Science All Hands Meeting*, 2004.

[57] D. Hollingsworth. The workflow reference model, 1.1. WfMC Specification, WfMC, January 1995. Available at: http://www.wfmc.org/index.php?option=com_docman&task=doc_download&gid=92&Itemid=72.

[58] M. Bell. *Service-Oriented Modeling (SOA): Service Analysis, Design, and Architecture*. Wiley, 2008.

[59] J. Bhagat, F. Tanoh, E. Nzuobontane, T. Laurent, J. Orlowski, M. Roos, K. Wolstencroft, S. Aleksejevs, R. Stevens, S. Pettifer, R. Lopez, and C.A. Gobel. BioCatalogue: a universal catalogue of web services for the life sciences. *Nucleic Acids Research*, 38(suppl. 2):W689–W694, 2010.

[60] C.A. Goble, J. Bhagat, S. Aleksejevs, D. Cruickshank, D. Michaelides, D. Newman, M. Borkum, S. Bechhofer, M. Roos, P. Li, et al. myExperiment: a repository and social network for the sharing of bioinformatics workflows. *Nucleic Acids Research*, 38(suppl. 2):W677–W682, 2010.

[61] C.J. Wroe, C.A. Goble, A. Goderis, P. Lord, S. Miles, J. Papay, P. Alper, and L. Moreau. Recycling workflows and services through discovery and reuse. *Concurrency and Computation: Practice and Experience*, 19(2):181–194, 2006.

[62] J. Wainer, M. Weske, G. Vossen, and C.B. Medeiros. Scientific workflow systems. In *Proceedings of the NSF Workshop on Workflow and Process Automation: State of the Art and Future Directions*. NSF, 1996.

[63] W.M. Johnston, J.R. Hanna, and R.J. Millar. Advances in dataflow programming languages. *ACM Computing Surveys*, 36(1):1–34, 2004.

[64] W. Van Der Aalst and K. Van Hee. *Workflow Management: Models, Methods, and Systems*. MIT Press, 2004.

[65] I.J. Taylor, E. Deelman, D.B. Gannon, and M. Shields. *Workflows for e-Science: Scientific Workflows for Grids*. Springer London, 2007.

[66] S. Bowers, B. Ludascher, A.H.H. Ngu, and T. Critchlow. Enabling scientific workflow reuse through structured composition of dataflow and control-flow. In *Proceedings of the 22nd IEEE International Conference on Data Engineering Workshops*, pages 70–70. IEEE Computer Society, 2006.

[67] S. Lu and J. Zhang. Collaborative scientific workflows supporting collaborative science. *International Journal of Business Process Integration and Management*, 5(2):185–199, 2011.

[68] S.L. Peyton Jones. *The Implementation of Functional Programming Languages*. Prentice-Hall, Inc., 1987.

[69] J. Gray, D.T. Liu, M. Nieto-Santisteban, A. Szalay, D.J. DeWitt, and G. Heber. Scientific data management in the coming decade. *ACM SIGMOD Record*, 34(4):34–41, 2005.

[70] U. Radetzki, U. Leser, S.C. Schulze-Rauschenbach, J. Zimmermann, J. Lüssem, T. Bode, and A.B. Cremers. Adapters, shims, and glue – service interoperability for in silico experiments. *Bioinformatics*, 22(9):1137–1143, 2006.

[71] E. Deelman and A. Chervenak. Data management challenges of data-intensive scientific workflows. In *Proceedings of the 8th IEEE International Symposium on Cluster Computing and the Grid*, pages 687–692. IEEE Computer Society, 2008.

[72] C. Hagen and G. Alonso. Exception handling in workflow management systems. *IEEE Transactions on Software Engineering*, 26(10):943–958, 2000.

[73] B. Clifford, I.T. Foster, J. Voeckler, M. Wilde, and Y. Zhao. Tracking provenance in a virtual data grid. *Concurrency and Computation: Practice and Experience*, 20(5):565–575, 2008.

[74] J. Golbeck. Weaving a web of trust. *Science*, 321(5896):1640–1641, 2008.

[75] K. Belhajjame, H. Deus, D. Garijo, G. Klyne, P. Missier, S. Soiland-Reyes, and S. Zednik. PROV model primer. W3C Working Draft, W3C, March 2012. Available at: http://www.w3.org/TR/2013/WD-prov-primer-20130312/.

[76] C.A. Goble. Position statement: Musings on provenance, workflow and (Semantic Web) annotations for bioinformatics. In *Proceedings of the Workshop on Data Derivation and Provenance*, 2002.

[77] J.A. Zachman. A framework for information systems architecture. *IBM Systems*, 26(3):276–292, 1987.

[78] T. Margaritopoulos, M. Margaritopoulos, I. Mavridis, and A. Manitsaris. A conceptual framework for metadata quality assessment. In *Proceedings of the 2008 International Conference on Dublin Core and Metadata Applications*, DCMI '08, pages 104–113. Dublin Core Metadata Initiative, 2008.

[79] M. Gamble and C.A. Goble. Quality, trust, and utility of scientific data on the Web: Towards a joint model. In *Proceedings of the International Conference on Web Science 2011*, pages 1–8, 2011.

[80] L. Huang, D.W. Walker, Y. Huang, and O.F. Rana. Dynamic web service selection for workflow optimisation. In *Proceedings of the 4th UK e-Science All Hands Meeting*, 2005.

[81] M. Greenwood, C.A. Goble, R.D. Stevens, J. Zhao, M. Addis, D. Marvin, L. Moreau, and T. Oinn. Provenance of e-science experiments – experience from bioinformatics. In *Proceedings of 2nd UK e-Science All Hands Meeting*, pages 223–226, 2003.

[82] S. Miles, P. Groth, M. Branco, and L. Moreau. The requirements of using provenance in e-science experiments. *Journal of Grid Computing*, 5(1):1–25, 2007.

[83] S. Miles, S.C. Wong, W. Fang, P. Groth, K. Zauner, and L. Moreau. Provenance-based validation of e-science experiments. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(1):28–38, 2007.

[84] F.C. Fang, R.G. Steen, and A. Casadevall. Misconduct accounts for the majority of retracted scientific publications. *Proceedings of the National Academy of Sciences*, 109(42):17028–17033, 2012.

[85] B. Borrell. Fraud rocks protein community. *Nature*, 462(7276):970, December 2009. Available at: http://www.nature.com/news/2009/091222/full/462970a.html.

[86] W.T.A. Harrison, J. Simpson, and M. Weil. Editorial. *Acta Crystallographica Section E*, 66(1):e1–e2, January 2010.

[87] L. Moreau, B. Ludäscher, I. Altintas, R.S. Barga, S. Bowers, S. Callahan, G. Chin, B. Clifford, S. Cohen, S. Cohen-Boulakia, et al. Special issue: The first provenance challenge. *Concurrency and Computation: Practice and Experience*, 20(5):409–418, 2008.

[88] L. Moreau and I.T. Foster, editors. *Provenance and Annotation of Data, International Provenance and Annotation Workshop, IPAW 2006, Chicago, IL, USA, May 3-5, 2006, Revised Selected Papers*, volume 4145 of *Lecture Notes in Computer Science*. Springer, 2006.

[89] L. Moreau, B. Clifford, J. Freire, J. Futrelle, Y. Gil, P. Groth, N. Kwasnikowska, S. Miles, P. Missier, J. Myers, et al. The open provenance model – core specification (v1.1). *Future Generation Computer Systems*, July 2010.

[90] J. Zhao. Open provenance model vocabulary specification. Technical report, October 2010. Available at: http://purl.org/net/opmv/ns-20101006.

[91] P. Groth and L. Moreau. An overview of the PROV family of documents. W3C Working Draft, W3C, March 2013. Available at: http://www.w3.org/TR/2013/WD-prov-overview-20130312/.

[92] L. Moreau and P. Missier. PROV-DM: The PROV data model. W3C Working Draft, W3C, March 2013. Available at: http://www.w3.org/TR/2013/PR-prov-dm-20130312/.

[93] D. Garijo and Y. Gil. Augmenting PROV with plans in P-PLAN: Scientific processes as linked data. In *Proceedings of the 2nd International Workshop on Linked Science*, 2012.

[94] L.N. Soldatova and R.D. King. An ontology of scientific experiments. *Journal of the Royal Society Interface*, 3(11):795–803, 2006.

[95] I. Niles and A. Pease. Towards a standard upper ontology. In *Proceedings of the International Conference on Formal Ontology in Information Systems*, pages 2–9. ACM, 2001.

[96] P. Panov, L.N. Soldatova, and S. Džeroski. Towards an ontology of data mining investigations. In *Proceedings of the 12th International Conference on Discovery Science*, DS '09, pages 257–271. Springer-Verlag, 2009.

[97] M. Courtot, W. Bug, F. Gibson, A.L. Lister, J. Malone, D. Schober, R. Brinkman, and A. Ruttenberg. The OWL of biomedical investigations. In *Proceedings of the OWLED Workshop on OWL: Experiences and Directions, collocated with the 7th International Semantic Web Conference*, 2008.

[98] B. Smith, P. Grenon, and L. Goldberg. Biodynamic ontology: Applying BFO in the biomedical domain. *Studies in Health and Technology Informatics*, 102:20–38, 2004.

[99] E.R. Cohen, T. Cvitaš, J.G. Frey, B. Holmström, K. Kuchitsu, R. Marquardt, I. Mills, F. Pavase, M. Quack, J. Stohner, H. L. Strauss, M. Takami, and A. J. Thor. *Quantities, Units and Symbols in Physical Chemistry.* Royal Society of Chemistry, 3rd edition, 2007.

[100] L. Lamport. *LaTeX : a document preparation system.* Addison-Wesley, 2nd edition, 1994.

[101] United Nations. *Globally Harmonized System of Classification and Labelling of Chemicals (GHS).* United Nations, 4th revised edition, 2011.

[102] European Union. New European Regulation (EC) No 1272/2008 on Classification, Labelling and Packaging of Substances and Mixtures (CLP Regulation). *Official Journal of the European Union*, December 2008.

[103] H.E. Pence and A. Williams. ChemSpider: An online chemical information resource. *Journal of Chemical Education*, 87(11):1123–1124, 2010.

[104] N. Adams, E.O. Cannon, and P. Murray-Rust. ChemAxiom - an ontological framework for chemistry in science. *Proceedings of the International Conference on Biomedical Ontology*, 1:15–18, 2009.

[105] R. Kidd. RSC and Southampton drive the chemical semantic web. Available at: http://www.chemspider.com/blog/rsc-publishing-and-southampton-university-drive-the-chemical-semantic-web.html, May 2011.

[106] Health and Safety Executive (HSE). *A step by step guide to COSHH assessment.* Health and Safety Guidance. HSE Books, 2004.

[107] K. Popper. The aim of science. *Ratio 1*, pages 24–35, 1957.

[108] K. Popper. *The Logic of Scientific Discovery.* New York: Basic Books, 1961.

[109] Nicholas Maxwell. A critique of popper's views on scientific method. *Philosophy of Science*, pages 131–152, 1972.

[110] B.C. Smith. *On the Origin of Objects.* The MIT Press, 1996.

[111] D. Davidson. *Essays on Actions and Events.* Clarendon Oxford, 1989.

[112] G.E.M. Anscombe. *Intention.* Harvard University Press, 1957.

[113] S. Hampshire. *Thought and action.* Chatto and Windus London, 1959.

[114] M. Borkum, S.J. Coles, and J.G. Frey. Integration of oreChem with the e-Crystals repository for crystal structures. In *Proceedings of the 9th UK e-Science All Hands Meeting*, 2010.

[115] S.J. Coles and L. Lyon. The eCrystals federation. In *Proceedings of the 3rd International Conference on Open Repositories*, April 2008.

[116] University of Alabama at Birmingham. UAB statement on protein data bank issues. Available at: http://main.uab.edu/Sites/reporter/articles/71570/, August 2009.

[117] G. Pólya. *How to Solve It.* Princeton University Press, 2nd edition, 1957.