Spring 3-23-2010

PROJECT MANAGEMENT PROCESS ONTOLOGIES: A PROOF OF CONCEPT

Robert Hughes
University of Brighton, r.t.hughes@brighton.ac.uk

Follow this and additional works at: http://aisel.aisnet.org/ukais2010

Recommended Citation
http://aisel.aisnet.org/ukais2010/30

This material is brought to you by the UK Academy for Information Systems at AIS Electronic Library (AISeL). It has been accepted for inclusion in UK Academy for Information Systems Conference Proceedings 2010 by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
PROJECT MANAGEMENT PROCESS
ONTOLOGIES: A PROOF OF CONCEPT

Robert T. Hughes
Department of Computing, Mathematical and Information Sciences, University of
Brighton, Watts Building, Moulsecoomb, Brighton BN2 4GJ UK.
Email: r.t.hughes@brighon.ac.uk

Abstract
Work is described that establishes the feasibility of representing project management processes as onto-
logies. The proof of concept is the creation of a prototype ontological model of the project
management processes prescribed by the PRINCE2® project management method, the de facto
standard sponsored by the United Kingdom government for public sector projects. The prototype was
created using the Protégé 4 ontological modelling tool which generates a representation of the
ontology as a collection of axioms expressed in OWL, a language developed to facilitate the
development of semantic web applications. The prototype confirmed the usefulness of ontologies in
modelling project management processes. Further potential applications have been identified such as
supporting process model tailoring, and checking the compliance of suppliers stated methods with
standards. However, the Protégé 4 ontological modelling environment would appear to present
difficulties for domain experts wishing to codify bodies of knowledge as its reasoning rules can be
counter-intuitive.

Keywords: PRINCE2, Protégé 4, project management, process model, ontology

1.0 Introduction
There is justified dissatisfaction with the frequent failures in IT development projects,
particularly in the public sector. For example, a report (Hendy, Reeves et al. 2005) on
the UK health service’s national programme for information technology (NPfIT)
noted a 30% failure rate for IT projects. One such IT project failure in the Wessex
Regional Health Authority led to a loss of £43 millions of public money. One
response has been guidelines and standards designed to ensure that project
participants follow perceived best practice in the conduct of projects. In particular,
the PRINCE2® project management method is recommended by the UK government
for use on public sector projects (OGC 2009a).

The problem of information overload for senior managers is well-documented
(e.g. Abbot 1999). The main PRINCE2® guide (OGC 2009a) is 327 pages long, and
is one of several sets of guidance which include the Gateway™ process (OGC
2009b), the MoR® risk management guidance for practitioners (OGC 2007a), and
programme management guidelines (OGC 2007b). The PRINCE2® content is
relatively circumscribed: it omits day-to-day project management techniques or development lifecycles with which an IT project manager would also be expected to be familiar.

A large number of written procedures incurs the risk that project progress is slowed because by unwieldy bureaucratic processes – an example of the danger of means/end inversion (Fitzgerald, Russo et al. 2003) – or that only lip service is paid to the standard – as evidenced by the frequent references to the PINO (‘PRINCE2 In Name Only’) method (Murray 2009).

To counter these risks, more lightweight or agile methods have been advocated – for example extreme programming (Beck and Andres 2005), Scrum (Schwaber 1997) and Atern (DSDM 2007). Regardless of the merits of these approaches, public servants and contractors in all professional fields still need to assure their masters – ultimately the general public – that they follow good practice and can provide evidence to this effect. One strategy is to use information technologies to hold knowledge bases of professional technical data and procedures in an easily accessible form. An example is the use of desktop computers to inform medical practitioners of relevant clinical procedures (Gunter 2005; Taylor 2006). An approach which has benefits in clinical situations might be usefully extended to IT and in particular to the management of IT implementation projects.

This paper explores the benefits of using ontological modelling – a technique widely used in biomedical environments – to map project development processes as an alternative to process modelling notations such as UML (Holt 2009), BPMN (White and Miers 2008) and Riva (Ould 2005). This is motivated by the possibility that ontological modelling could facilitate:

- The establishment of internal consistency within process descriptions;
- Speedier access to the processes that have a bearing on a situation;
- A clearer means by which standard project management procedures can be tailored to meet the circumstances of a project;
- The alignment of project management processes with other, complementary, processes such as those used to guide software development;
- The identification of appropriate information structures and systems to hold operational project data, such as the details of activity timings and durations and resource allocations.

The research question addressed in this paper is whether project management processes can be accurately represented by an ontological model. To establish this, a prototype ontology capturing PRINCE® processes was developed using the Protégé 4 ontology modelling tool - see http://protege.stanford.edu for details of this tool. A
subsidiary research issue was the extent of the possible benefits of carrying out such
modelling.

The remainder of the paper is structured as follows. Section 2 provides an
overview of PRINCE2® as a methodology, setting this in the context of the
community of practice within which it is embedded. Section 3 presents a background
to ontological modelling, interest in which has been generated by the movement
towards the development of the ‘Semantic Web’. Section 4 explains how the
prototype PRINCE® ontology was implemented using Protégé 4, describes some of
the challenges of modelling PRINCE® and discusses some of the initial – but not yet
conclusive – validation of the resulting model. The final section examines the benefits
of the use of ontologies for project management and some relevant future research
directions.

2.0 The PRINCE2® project management method

2.1 The development of the method

One purpose of this section is to justify the choice of PRINCE2 as a representative
project management process to be modelled in a prototype. The second is to explain
the general nature of the target knowledge base. Some ontologies describe a
(relatively) objective knowledge base, such as the laws of physics where, generally,
disputes about interpretation can be resolved by examination of a physical state of
affairs. PRINCE2, however, is a political and social entity describing what should be
rather than what is, and this introduces its own challenges.

The original PRINCE (‘Projects IN Controlled Environments’) was
published in 1989 by the Central Computer and Telecommunications Agency
(CCTA), a UK government body attached to HM Treasury and responsible for
government-wide IT standards. PRINCE was essentially a set of procedures regarded
as best practice for managing IT projects. A major innovation at the time was its focus
on the management of the products, both deliverable and intermediate, of a project,
rather than its activities.

PRINCE2® was a wide-ranging update in 1996 which broadened the
method’s scope beyond IT to all types of project. In practice, the main use of
PRINCE2® has broadly been in ‘business change’. Most organizational IS/IT projects
involve changes to the business, and, conversely, what are primarily business changes
are likely to have IS/IT implications. The latest update to the method, identified as
PRINCE2® 2009, was launched on 16th June 2009 by the Office of Government Commerce (OGC), the successor to the CCTA. This version is the basis for the work described in this paper.

2.2 PRINCE2 as a community of practice

A project methodology is a collection of procedures, techniques, tools and documentation which help project participants in the implementation of projects. While this paper focuses on PRINCE® as a theoretical construct that can be modelled using rigorous mathematically based techniques, it is recognised that methodologies are the product of social construction. Methodologies that are more than merely academic products will have a ‘community of practice’ (Wenger 1998) that shares the experience of using the methodology’s practices. The methodology-in-use will vary from the methodology-as-proposed as users make their own interpretations and adjustments to suit personal inclinations and local circumstances. A new methodology is a type of innovation and its uptake will be influenced by the same factors that affect all innovations, such as its fit with existing technologies and practices, the degree of irreversibility of investment, sponsorship and expectations as to the future general adoption (Fichman and Kemerer 1993)

PRINCE2® clearly benefits from the sponsorship of the UK government: this supports the expectation that the method will continue in use for public sector projects, and thus there will continue to be maintenance and the availability of the resources, expertise and materials to support the method. According to the OGC (2009a), PRINCE2® is used in 150 countries and by 20,000 organizations. The increasing adoption of PRINCE2® outside the UK is illustrated by the choice of the method by the French Finance Ministry for its modernization programme (OGC 2009c). In 2009, 131,000 PRINCE2® examinations (see below) were taken in nine different languages.

One characteristic of a community of practice is the means by new members are inducted. With PRINCE2®, individuals are accredited as PRINCE2® practitioners. Accreditation has been devolved by the OGC to a commercial organization, the APM Group which originally grew out the Association of Project Management (APM) but is now a separate concern. To gain accreditation, individuals take examinations at two levels, foundation which is based on multiple choice questions and the practitioner I level which uses an ‘objective test’. In order to retain
practitioner status an existing qualification holder must undertake a one hour, open book, ‘objective test’ every five years.

Passing these tests requires recall and recognition rather than a demonstration of practical skills in project management. PRINCE2® documentation does not claim this assessment process certifies the general management competence of individuals. At foundation level the aim is to assess whether an individual could ‘act as an informed member of a project management team using the PRINCE2 method within a project environment supporting PRINCE2’ (APM Group Limited 2010). The emphasis is on demonstrating an understanding of the method. At practitioner level, candidates are assessed as to whether ‘they could apply PRINCE2 to the running and managing of a non-complex project within an environment supporting PRINCE2’ (APM Group Limited 2010).

PRINCE2® documentation also makes clear the limited scope of the approach (OGC 2009a pages 6-7). Firstly, PRINCE2® describes a generic project process and does not cater for industry specific aspects of particular projects, so IS/IT practitioners would need to supplement PRINCE2® with, for example, the specification of an IS/IT project lifecycle. Secondly, PRINCE2® does not cover the practical techniques in standard project planning and control, such as activity networks, effort and duration planning, resource allocation, and earned value analysis. Finally, it does not cover ‘leadership, motivational and other interpersonal skills’.

PRINCE2® documentation also states ‘PRINCE2® provides a framework of what needs to be done, by whom and by when. The [body of knowledge] provides a range of techniques of how those things can be done’, emphasising that PRINCE2® is not a body of knowledge (BOK) (OGC 2009a pages 230-231). PRINCE2® is thus characterized as a set of integrated principles, processes and themes. Where appropriate, it calls upon complementary BOKs to provide detailed guidance for executing particular activities. As will be seen, the structured and integrated nature of PRINCE2® lends itself to ontological modeling. The use of multiple choice and objective testing in assessing PRINCE2® expertise would not be feasible without a high degree of precision and a relative lack of ambiguity in its definition which also suggests ease of conversion to an ontology. Ontologies often make use of existing complementary ontologies for support which conveniently fits with PRINCE2®’s use of supporting BOKs.
In the current work the choice of PRINCE2® as a representative project management approach is justified by its widespread adoption. In addition, the adoption of PRINCE2® has financial implications which justifies research exploring how the information contained within PRINCE2® texts can be accessed more effectively. Candidates for its examinations in the UK (or their employers) pay currently pay £200 at foundation level and £370 at practitioner level, while the three-yearly re-registration costs £145. Most candidates would attend accredited preparatory courses that could cost up to £1600 for a five day course (but this usually includes the examination fee). Organizations may also employ consultants to provide further guidance on PRINCE2®, particularly on tailoring and embedding PRINCE2® in the context of their organization. The method requires project assurance and project support roles, and employing and servicing staff to carry out these roles would add to cost. The greater proportion of the financial outlays above can be seen as relating to information and knowledge management tasks, and thus the use of computer-supported ontologies that have been found to be useful in other fields is worth examining in this one.

3.0 Ontologies and Protégé

3.1 What is an ontology?

In this section the concept of ontology is introduced. Firstly, the concept of the term ‘ontology’ in computing and information science is distinguished from its traditional use by philosophers. The characteristics of an ontology (in the computing sense) are then explored. The logical structure and content of such an ontology can be translated into a machine format in various ways, and an approach using the Protégé 4 tool will presently be used to explain the general principles behind the representation of ontologies.

The original meaning of ‘ontology’ is as the philosophy of being. The question of the nature of existence has preoccupied philosophers since at least Parmenides of Elea in the 5th and 4th centuries through Aristotle to the present era (Gómez-Pérez, Fernandez-López et al. 2004) – for example, Sartre’s Being and Time is subtitled ‘An essay on phenomenological ontology’.

In computer science, ‘ontology’ originally described a representation of the part of the ‘real world’ that supported a problem-solving artificial intelligence (AI) application. The early development of ontologies is often associated with the work of
Gruber (Gruber 1993; Gruber 1995) and Guarino (Guarino 1995). In AI, an ontology is analogous to the data models with which IS practitioners will be familiar. Ontology modelling and data modelling share many concepts and techniques. It could be that some of the differences between the two can be traced to the two disciplines being developed in parallel but independently of one another. More fundamentally, ontology modeling often attempts the analysis of less precisely defined and less rigidly structured information than data modelling.

3.2 The influence of the Semantic Web

The interest in ontological modeling was boosted by the drive to improve the usability and reliability of the World Wide Web (WWW) through the development of a ‘semantic web’. Tim Berners-Lee, the father of the WWW, has been a major influence on semantic web developments. In his seminal paper (Berners-Lee 2002) proposing a programme of work on the Semantic Web, there is an emphasis on the World Wide Web as a facilitator of collaborative projects: ‘The first goal was to enable people to work together better….The idea was that by building a hypertext Web, a group of whatever size would force itself to use a common vocabulary, to overcome its misunderstandings, and at the same time have a running model – in the Web – of its plans and reasons’ (page xiii). This needed ‘a map, in cyberspace, of all the dependencies and relationships that defined how a project was going’ (page xiv). This required the modeling of metadata, information about information, and the ideal repository for this seemed to be an ontology.

3.3 Ontological commitment

An ontology – what Gruber called ‘an explicit representation of a conceptualization’ – is essentially designed rather than discovered. Different individuals have different perceptions of the same underlying reality. For example, professionals focus on those elements of an object which are the subject of their specialism. A successful ontology requires a shared ‘ontological commitment’ by those – often a community of practice - who will use the ontology. This is an agreement to see a subject domain in a certain way and to use a common terminology to communicate about the domain. Thus PRINCE2® can be seen as a project management ontology to which PRINCE2 practitioners have an ontological commitment. This commitment is not unconditional and might be a pragmatic temporary concession in order to get work done.

The work described in this paper has not attempted a web-based implementation but the ontology modelling tool, Protégé 4, generates ontological
representations in OWL (the Web Ontology Language – the fact that the acronym is OWL rather that WOL is deliberate, if confusing) which has been designed to facilitate representation on the World Wide Web.

3.4 Protégé

The underlying structure of data in Protégé-OWL can be visualized as a set of triples in a subject-predicate-object form where the subject and object are classes and the predicate is a property that links them, for example:

(Employee, isProjectManagerOf, Project)

Note the naming convention whereby class names have a capital initial letter and the property starts with a lower case character. The relationship identified by the predicate can either be ‘existential’ where an instance of the Employee class could be linked to several Projects, or ‘universal’ where it can only be linked to one.

A special type of relationship is based on subsumption where one class subsumes one or more subclasses. For example, the class Person could subsume the subclass Employee, as well as, say, Customer and Contractor. Subclasses could subsume sub-subclasses. For example, the subclass Employee might subsume Manager. A subclass inherits all the properties of the superclass by which it is subsumed.

Classes can also be distinguished from instances of the class. A class is a generic description of an entity such as Employee, and there could be several instances of this class, such as the ones with the names ‘Joan Smith’ and ‘John Brown’. For further information about Protégé-OWL, see Horridge (2009).

These basic building blocks can be amalgamated to create complex knowledge structures. These assemblies of assertions describing an ontology and expressed as triples can be interrogated by tools which can identify new classes not originally asserted, but which can be inferred from previous assertions. For example, there might be two additional assertions:

(Project, usesMethodology, Methodology)
(Employee, isAccreditedIn, Methodology).

If there is a requirement that the ‘ProjectManager’ for a project using PRINCE2® must be accredited in PRINCE2®, a class of Employees qualified to manage the PRINCE2® Projects can be inferred. If ‘Project A’ is an instance of Project and is linked to ‘PRINCE2®’ via the usesMethodology property, and if ‘Joan Smith’ is an instance of Employee and is linked via the isAccreditedIn property to ‘PRINCE2®’ as
Methodology then it is inferred that she is eligible for the role of ProjectManager of ‘Project A.’

The version of Protégé used for the PRINCE2® prototype ontology used a version of OWL compatible with a mathematical representation called description logic or DL (Baader, Calvanese et al. 2003). The DL notation is usually hidden by the Protégé interface, but certain assumptions that characterize DL can be traps for non-specialist users. This is because DL adheres to what is called the ‘open world assumption’ whereby something is not assumed not to exist unless there is an explicit assertion that it does not exist.

4.0 Building the Prinny Ontology
The prototype was called ‘Prinny’ after the Prince Regent, Prince William, who had a strong association with Brighton. In this section the process of building the Prinny prototype ontology is outlined. A description of the structure of the resulting ontology then follows. While constructing the prototype some problems were experienced in interpreting the PRINCE2® textual description upon which it was based and these are discussed. There is also some discussion of the problems of using Protégé. Finally the initial validation of the model is touched upon.

4.1 Scope of the Prinny prototype ontology
A prototype is built in order to gain some knowledge. The cost of building the prototype can be seen as the cost of acquiring that knowledge. When planning a prototype, an informal cost-benefit calculation may be involved, so that unnecessary effort is not expended on the perfection of a prototype that does not add significantly to the knowledge generated (Hughes and Cotterell 2009).

Application prototypes can be categorized as:

- **vertical** where only a subset of the functionality needed in the final application is prototyped but the detail of the functionality for that proportion is as close as possible to that delivered on completion, or

- **horizontal** where the full range of functions in the application are prototyped but not all the details of the functions are completed.

Given these broad alternatives a decision had to be made as to the scope of the ontological model. To understand the nature of the decisions about scope, a view of the structure of PRINCE2® is needed.

For the purposes of this paper, PRINCE2® can be divided into:
a) Seven principles – see Table 1
b) Seven themes – See Table 2.
c) Seven top-level Processes – See Table 3
d) Advice on tailoring PRINCE2® to the project environment
e) Product description outlines
f) Roles and responsibilities

The seven principles are presented as the key criteria by which a project can be judged as adhering to the PRINCE2® standard – see Table 1.

<table>
<thead>
<tr>
<th></th>
<th>Continued business justification throughout the project</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Learning from experience</td>
</tr>
<tr>
<td>2</td>
<td>Definition of roles and responsibilities</td>
</tr>
<tr>
<td>3</td>
<td>Division of a project into one or more stages for management purposes</td>
</tr>
<tr>
<td>4</td>
<td>Management by exception</td>
</tr>
<tr>
<td>5</td>
<td>Focus on products</td>
</tr>
<tr>
<td>6</td>
<td>Tailoring PRINCE2® processes to fit the environment of a project</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 1</th>
<th>The seven PRINCE2® principles</th>
</tr>
</thead>
</table>

The seven themes identify aspects of project management that must be addressed continuously throughout the project – see Table 2.

<table>
<thead>
<tr>
<th></th>
<th>The business case</th>
<th>5. Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Organization</td>
<td>6. Change</td>
</tr>
<tr>
<td>2</td>
<td>Quality</td>
<td>7. Progress</td>
</tr>
<tr>
<td>3</td>
<td>Plans</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2</th>
<th>The seven PRINCE2® themes</th>
</tr>
</thead>
</table>

In PRINCE2® the actions to address the concerns explored by the themes in Table 2 are presented in a number of specific processes – see Table 3. One process, for example, *Starting up a project*, could describe some actions relating to several themes, and vice versa. Each theme identifies the responsibilities of defined project management roles (corporate/programme, executive, senior supplier, senior user, project manager, team manager, project assurance and project support) for that theme. The descriptions of the processes also allocate responsibilities to each role for each action within a process. In theory, by cross-referencing the role responsibilities shared by themes and process actions, it should be possible to cross-reference themes and processes – but there is no explicit mapping. In some cases the indirect linking is straightforward: for example, within the organization theme, the Project Manager role is allocated the responsibility for preparing the communication strategy within the *Initiating a Project* process. In other cases, a responsibility is more general: for
example, the Team Manager role within the organization theme has a responsibility to ‘manage team members’ which is not easily linked to specific PRINCE2® actions.

<table>
<thead>
<tr>
<th></th>
<th>PRINCE2® processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Starting up a project</td>
</tr>
<tr>
<td>2.</td>
<td>Directing a project</td>
</tr>
<tr>
<td>3.</td>
<td>Initiating a project</td>
</tr>
<tr>
<td>4.</td>
<td>Controlling a stage</td>
</tr>
<tr>
<td>5.</td>
<td>Managing project delivery</td>
</tr>
<tr>
<td>6.</td>
<td>Managing stage boundary</td>
</tr>
<tr>
<td>7.</td>
<td>Closing a project</td>
</tr>
</tbody>
</table>

Table 3

A newcomer to PRINCE2® may find identifying all the duties and responsibilities of a role difficult as they are defined in three different places: under themes, processes and roles. There can be differences in the three sources: for example, in the appendix describing the responsibilities of the project manager, no mention is made of quality management, while quality management responsibilities are clearly identified for the project manager in the section on the quality theme.

This illustrates a benefit of machine-based ontological models. Such a model could identify role responsibilities in one place, and then link them to roles, processes and themes. This information can then be extracted and formatted for different purposes while maintaining its internal consistency.

From the point of view of prototype ontology construction, the lack of precision in the PRINCE2® text in relation to roles and responsibilities meant that too many assumptions and interpretations would be needed leading to an ontology model at variance with the PRINCE2® text, or, alternatively, the ontology would have to reflect the inconsistencies of the text. Neither alternative seemed attractive.

Processes were more clearly defined and structured: processes were broken down into activities and each activity had descriptive text explaining the recommended actions to be carried out within that activity. Diagrams identified the products used by each activity and tables identified the roles responsible for the creation or modification of each management product. This area of PRINCE2® was selected for building as an ontology as it seemed to lack ambiguity.
The following classes were chosen for the ontology:

- **Process** – see Table 3
- **Activity**: a step in a process
- **Interaction**: this is not a PRINCE2® term and refers to the interaction between particular roles and the products they create and modify within an activity.
- **Product**: note that these refer almost exclusively to management products used to control the project, not the actual deliverables of the project. No attempt was made to record in the ontology any details describing products, such as purpose, composition, derivation, format and presentation, and quality criteria, as this would basically be simply transcribing text.
- **Trigger**: this is where one activity sets a trigger which causes another activity, normally in a different process, to start.
- **Role**, that is a set of responsibilities: note that different aspects of the same role could be carried out by different individuals or even that one individual, on a smaller project, could carry out more than one role.
Figure 1 above shows the top-level classes and the object properties that linked them and is effectively a process metamodel. Although the design is based specifically on the structure of PRINCE2® process model, its general structure means it could be applied to other process models. Each of the classes had a set of subclasses defined, so that, for example, for the class Process, the seven processes in Table 3 were specified as subclasses. 40 distinct types of Activity were coded and 144 Interactions were coded. 45 types of Product were identified as being either created or updated by Activity subclasses – a Product could be updated by more than one Activity.

While the prototype only modelled a subset of the candidate classes identified in the text, all the instances of the selected classes were modelled – hence the prototype can be seen as a horizontal. This approach was selected because of the integrated nature of PRINCE2®. A partial model of the selected classes would have lacked value as the cross-references across processes would have been incomplete.

Description logic distinguishes two parts of a knowledge base, a TBox and an ABox. The TBox - or Taxonomy – describes the metadata, or information about the structure of the data held in the ontology. This information will tend to be stable over time. The ABox contains information about individual instances of the types of data in the real world corresponding to each element in the TBox. Currently, Prinny contains only TBox information. ABox information would be added if Prinny were to be used as effectively a project support tool holding items of data, such as project plans or entries in a risk register relating to individual projects, rather than just details of requirements for such data that potentially apply to all projects.

It has been noted above that the classes in Figure 1 can be seen as constituting a metamodel. Interestingly, the generic metamodel and the process-specific model are held in the same format and can be stored in the same knowledge base. This provides opportunities of ontology mapping and merging that will be discussed in more detail in Section 5.

4.2 Populating the Prinny framework

When designing an ontological metamodel, the modeller has to make an assumption that the content that will populate the structure will be compatible with that metamodel framework. Where the source information used is text-based, there is a risk that there will be anomalies that will not fit the framework.

The PRINCE2® text sets out the format for describing management activities (OGC 2009a). A process is a set of activities that has input documents, actions to
generate outputs (which can be revisions of the input documents), and the roles responsible for the actions on each product. In some cases, however, in the descriptions of individual activities, actions appear to be allocated to input documents rather than outputs. Where an action is to approve an input product there is no problem, as this can be seen as updating the status of the product and effectively creating a new baseline product ready for processing by the next activity. However, in other cases, where the action on the input document is ‘Respond’, ‘Inspect’ or ‘Obtain’, for example, the precise nature of the transformation that might create an output is unclear.

There also appeared to be a lack of consistency with the creation of Triggers. In some cases, the setting of a Trigger was treated as an output for which responsibilities were allocated. In other cases, this was left undefined.

The execution of some activities was initiated by a trigger, but most were not. The PRINCE2® text contained diagrams with arrows between activities, but it was not clear whether these indicated flows of data between activities or a temporal sequencing where one activity could only start when another had been completed. It could be surmised that most activities would be triggered when a key input product was ready for processing, but products with this property were not explicitly identified. The identification of pre-conditions for each activity would have been helpful.

When the responsibilities for actions which created or updated a management product were allocated some of these were indicated as executed outside the current activity. For example, a planning document might be created and reviewed in one activity but be approved elsewhere. The PRINCE2® text did not specify where the other activity was or whether the approval was done outside of the framework of PRINCE2® processes and activities. This created ambiguity as to the nature of the links between different activities, particularly where one activity depended on products created by other activities and the place where these products would be approved was not clear.

### 4.3 Experience of Protégé 4

It would be fair to say that Protégé 4 was designed for use for researchers rather than practitioners. Much of the published work on it relates to the development of the tool itself and the development of plug-ins which extend Protégé 4’s capabilities.
The developmental nature of Protégé 4 and its underpinning foundation in description logic means that the interface with the tool for the new naïve user is not as friendly as, say, the use of a desktop data management tool such as Microsoft Access. With ontological models there is no rigid division between data and metadata. Because of this, what are seemingly obvious input errors - such as assigning the wrong type of property to a class – which would be picked up on input with conventional data management systems are not rejected. Instead, classifiers, analogous to code compilers, are run after the data has been incorporated in the model. A single, trivial, error can cause the classifier to highlight consequential errors that can cause most or all of the assertions in the ontological model to be flagged in red. Identifying the precise source of problems can be time consuming. As Parsia, Sirin et al. (2005) note: ‘The tool has told [the users] that there is a problem, but given no help in fixing it. This has two negative consequences: either developers specify their concepts to “avoid” error (at least, to avoid “fatal” error) or they give up ontologies altogether’.

This was not simply a matter of interface design: the underpinning description logic processing was often counter-intuitive. For example, it was not possible to query directly whether there were any instances of an Activity which accessed Product instances but did not create or update new ones. The rationale seems to be that the open world assumption means that just because no Product is currently recorded as being created and updated by an Activity, it might be because current information was incomplete rather than there being certainty there was no Product update or creation.

4.4 Initial validation of Prinny

Validation of Prinny was based firstly on ensuring that classifier software tools found no inconsistencies in the structures that had been created. Queries were also run that generated output that could be checked back again the PRINCE2® text. Other queries checked for the internal consistency in terms of the known characteristics of the ‘reality’ being modelled. For example, every Trigger was checked to see that there were both an Activity which could set it and at least one Activity that it fired. In fact, some valid cases were found where one of these elements was missing: a Trigger might be set by an event in the environment of PRINCE2®, or PRINCE2® could trigger an event in its environment such as the start of operation of a system that a project had just delivered.
Another such ‘reality check’ was that each Product had Activities that created and used it. Some anomalies were found. These could be coding errors, gaps in PRINCE2®, or situations similar to the ones identified for Triggers where Products were received or delivered over the PRINCE2® system boundary.

Validation and refinement of the basic Prinny model is currently ongoing.

5. **Some conclusions and future directions**

5.1 **Feasibility of ontological modelling of processes**

The work described in this paper demonstrates the technical feasibility of implementing a description of PRINCE2® management processes as an ontological model using Protégé 4. It could, however, be argued that this has been made easier by ‘cherry picking’ those aspects of PRINCE2® most susceptible to representation as an ontology.

It is conceded that this exercise was harvesting low-hanging fruit in its selection of the features of PRINCE2® to model. However, the basic subject-predicate-object structure of the modelling language suggests that much, if not all, the remaining PRINCE2® text could be modelled. The issue is that modelling is easiest where the source text is most structured. Structuredness is where a relatively small number of classes have a large numbers of instances. Where the content has a large number of classes but few instances, analysis and model-building will be more demanding, and the benefits will be smaller.

The work on Prinny has demonstrated the value of the ontological modelling of process models in ensuring that such models are internally consistent.

5.2 **Future directions**

The following possibilities are opened up by applying ontological modelling to process models

*Ontology matching*. This is where ontologies are compared to identify similarities and differences (Kalfoglou and Schorlemmer 2003; Choi, Song et al. 2006; Shvaiko and Euzenat 2008). An example application is where a price comparison website needs to match catalogues from different suppliers in order to present the user with a unified view of the market. Suppliers may vary in their coverage of available products and in the way that they classify them. In the case of process models, an example would be where contractors from a non-UK country bid for work in the UK public sector. The contractors may already use a local non-PRINCE2 project management model.
Ontologies of the two process models could be matched to see whether they are equivalent.

**Ontology merging.** In the above example, it might be that the ontologies are broadly equivalent and in some cases complementary, that is, that one ontology covers some areas that the other does not and vice versa. It might be possible to construct a merged ontology showing how the two process models could interact. Future possible work to take forward the modelling described in this paper includes a plan to create an ontology for Atern, the agile development approach, and then seek to merge it with Prinny. One way of merging would be to analyse Atern using the metamodel shown in Figure 1. Subclasses could be set up which added specific Atern Products and Activities. In some cases some of these subclasses might be found to be equivalent to existing ones in the PRINCE2® ontology.

**Ontology tailoring** The PRINCE2® 2009 manual (OGC 2009a) emphasises the need for the appropriate tailoring of PRINCE2®, echoing a broader concern for the need to be able to adapt a core of standard processes in order to optimize their use in a particular context. For example, a project might be a part of a larger programme and as a consequence some aspects of its management might be subsumed by programme level processes and can thus be deleted at project level. The PRINCE2® 2009 manual states that PRINCE2® activities have to be retained regardless of the scope and context of the project, but that roles and products can be adapted. In some cases, for example, products can be amalgamated. A standard PRINCE2® ontology could be modified to take account of such tailoring and the consistency of the new version of the ontology could be checked. For example, in some cases merging products may mean some activities can be merged, but this may not always be the case.

### 5.3 Ontology modelling tools and techniques

A potential obstacle to the ontological modelling of process models is the lack of usability of at least some modelling tools and the counter-intuitive reasoning rules of an underpinning framework such as description logic. It is expecting a lot for developers to be both experts in the domain being modelled and as well as the intricacies of the modelling tools and representations. One approach (Rector, Wroe et al. 2001) is to divide developers into different groups with distinct specialist roles, some more domain-oriented and some who focus on the technical implementation. Another approach is to use methods that have been tried and tested in allied fields, such as the use of UML, to model ontologies (De Nicola, Missikoff et al. 2009).
Clearly, the process of creating ontologies of process models needs a process model itself.

**Acknowledgements**

I would like to thank Charlie Aitken, David Purves and Karl Cox for reviewing this paper and making many helpful suggestions for correction and improvement. The work was funded by the University of Brighton research sabbatical scheme.

**References**


