

# Balancing System Expressivity and User Cognitive Load in Semantically Enhanced Policy Modelling

Christos Tsarouchis, Declan O'Sullivan, and David Lewis

Knowledge & Data Engineering Group (KDEG),  
School of Computer Science & Statistics,  
Trinity College, Dublin, Ireland  
{tsaroucc,Declan.OSullivan,Dave.Lewis}@cs.tcd.ie

**Abstract.** Policy Engineering is the process of authoring management policies, detecting and resolving policy conflicts and revising existing policies to accommodate changing resources, business goals and business processes. Policy authoring involves developing a policy rule base populated with policies that specify where actions specified on behalf of subjects may or should be performed on targets (resources). Policy engineering can be a daunting process in terms of complexity of the subjects, targets and actions used in rules, and the potential for conflicting policy rules to be committed to the rule base. Semantic modelling of rule elements can help automate the detection of such conflicts, but the additional layers of abstractions may themselves add to the complexity faced by the policy author. In this paper, we aim to assist the policy author by increasing the system expressivity with semantics, while at the same time minimizing the perceived cognitive load due to additional model complexity. We build on work with abstractions aimed at achieving this goal in the modelling of organisational grouping for subjects of policy-rules, which supports specific authority-based group abstractions to ease the maintenance of the subject model in the face of frequent organisational change. In this paper, we study this approach as used in combination with description logic based modelling of target semantics plus logic programming assertions across subject, targets and actions. This is performed through detailed analysis for policy authoring deliberations observed in a user evaluation of these modelling techniques.

**Keywords:** Policy engineering, semantic modelling, description logic semantics, policy rule semantics.

## 1 Introduction

Policy Engineering is the process of authoring policies, detecting and resolving policy conflicts and revising existing policies to accommodate changing resources, business goals and business processes [1]. Policy engineering can be a daunting process in terms of complexity of the subjects, targets and actions used in rules, and the potential for conflicting policy rules to be committed to the rule base. The policy author's cognitive load [2] can be stressed in an environment of fluid organizational structure and changing resources, which are the targets for policy enforcement.

In this paper, we aim to assist the policy author by increasing the system expressivity with semantics, while at the same time minimizing the perceived cognitive load due to additional model complexity. Semantic modelling of rule elements can help automate the detection of system conflicts, but the additional layers of abstractions may themselves add to the complexity faced by the policy author.

Other research initiatives have attempted to solve such problems by combining the efficiency and the expressive power of rule-based systems with the semantic richness found in Description Logics [3]. For instance, a rule-based system can express variables and causality relationships whereas ontologies on the other hand, can represent complex relationships between entities, such as the relationships of equivalence and disjoint, in a taxonomic manner that could simplify maintainability.

In this paper, for the modelling of policy subjects, the rule-based Community-based Policy management (CBPM) [4] will be used. This integration falls into the category of combining Description Logics (DL) [5] and Logic Programs (LP) [6].

The structure of this paper is as follows: for the remainder of section 1, the benefits of semantically enhanced policy subjects and targets, as well as the integration of CBPM with DL, and the resulting syntax will be presented. In section 2 a case study based on the semantic encoding of the Trinity College Statutes, together with some examples, will be shown. An intervention analysis as part of a user trial series is presented in section 2.2. Section 3 discusses the paper findings, while in section 4, related and future work will be presented.

## 1.1 Semantically Enhanced Policy Subjects and Targets

One of the requirements in policy engineering is the need to have an accurate and up-to-date view of the managed resources as well as of the actions that can be performed on these resources. This is particularly challenging in the case when not only these resources are frequently changing but also when their corresponding actions change as well.

This can be achieved by having an up-to-date snapshot of the resources and their corresponding actions in question into a structured, manageable and easy to query knowledge base (KB). This KB can be represented using ontological representations. This is because the use of ontologies can provide a centralized form of knowledge aggregation, when even complex relationships can be described and queried. Using OWL DL [7] for instance, the “equivalent” and “disjoint” relationships between resources and actions can be described. An example application that showcases such a system is presented in [8].

When modelling organizational structures, the flow of authority between subjects and targets needs to be modelled and preserved. Large organizations have decision-making groups, which are usually hierarchical in nature and often delegate the decision authority to other groups lower in the hierarchy. To model such concepts the use of the ontological class-subclass relationship is not adequate. For example, a company's Director has authority over the Project Lead, but the latter is not a subclass of the former. In reality however, decision-making groups are more fluid and overlapping than the classic hierarchical organizational models that underpin role-based schemes. For this purpose, this research will use the Community-based Policy Management (CBPM) [4] framework to model policy subjects, which has been shown

to model the organizational aspects of policy subject models in a way that makes the resulting policy base more robust to organizational change [9].

CBPM uses the notion of Community as the primary grouping abstraction with the aim of allowing groups within the organization itself to define communities to naturally reflect the changing nature of decision making (i.e. policy setting) authority. Communities towards the top of the hierarchy have the wider membership and more general function, while those toward the bottom have more narrow membership and more specific function. The hierarchy is designed to support agility and autonomy, allowing new sub-communities to be formed and encouraging the delegation of decision making authority as far down the hierarchy as possible.

Using CBPM enables us to have a powerful permissions scheme in place, since the flow of authority from one community to another is explicitly defined. This scheme grants or denies permission to perform actions on resources and it can also be used to define obligations.

In this paper we therefore aim to investigate how different combinations of DL and LP impact on the overall system expressiveness and on the cognitive load placed upon the requirements engineer, while aiming for a similar level of policy engineering robustness for target models, which CBPM delivers for subject models.

We will now present in more detail, the inner workings of the integration between CBPM and OWL DL.

## 1.2 CBPM and OWL DL Integration

The type of integration between CBPM and OWL DL presented in this paper, is of the hybrid type [3], which means that there is no creation of a new logical language. The resulting framework is called CBPM+DL. CBPM+DL is formed as follows: a hybrid knowledge base  $K = (S,R)$  is a finite set  $S$  of DL axioms in the ontology language  $S$  and a finite set of hybrid rules  $R$  over  $R$  and  $S$ , including non-DL atoms. DL atoms are classes, properties, individuals. This definition follows the hybrid integration definition presented in [3].

Since the CBPM+DL method follows this hybrid approach, the semantics entailed in the hybrid language are derived from the semantics of its constituent components, namely CBPM and OWL DL. A hybrid rule  $r$  is obtained from the CBPM rule  $r'$ , by adding DL constraints in the body. The latter are in fact additional constraints on the use of the rule  $r'$ , which have to be satisfied so that the rule  $r'$  is fired.

### B CBPM+DL Syntax

The syntax of CBPM+DL is formed by extending the existing CBPM syntax with ontological predicates. A rule rendering of the existing CBPM syntax will now be presented, in which a rule has the form: antecedent  $\rightarrow$  consequent

The antecedent is a conjunction of atoms written  $a_1 \wedge a_2 \wedge \dots \wedge a_n$ . Variables are indicated using a question mark as a prefix (e.g. ?x).

#### 1) *Community Membership Rule*

The membership rule defines the restrictions, which need to be satisfied, so that a person  $P$  is a member of Community  $C$ .

Property is of type ontological object property restriction. This means that user-specific properties can be created.

2) *Authorization Rule*

Authorization is about granting or denying access (negative authorization) to a subject to perform an action on a resource. In this research we use the notion of resource authority, which is introduced in CBPM. A resource authority specifies what action on which resource, access is sought for.

The resource authority is defined as follows:

$\text{Action}(?a) \wedge \text{Resource}(?r) \rightarrow \text{ResourceAuthority}(?ra)$

The resource authority is used as a predicate in the authorization rule:

$\text{comm}(?comm) \wedge \text{isa}(?x, ?y) \rightarrow \text{posAuth}(?comm, ?ra)$

The  $\text{isa}(?x, ?y)$  denotes the OWL is-a relationship, and in this case it means that  $x$  is a subclass of  $y$ . This rule can be read that as long as the "isa" relationship is satisfied, the community  $\text{comm}$ , is authorized to perform an action on a resource, as specified in the resource authority restriction.

3) *Implies authority rule*

The implies authority relationship describes the authority which is exercised by a Community, an Action, and a Resource on a Community', an Action' and a Resource' respectively, which are located lower in the hierarchy of authority.

$\text{Community ImpliesAuthorityOver Community}'$

$\text{Action ImpliesAuthorityOver Action}'$

$\text{Resource ImpliesAuthorityOver Resource}'$

': lower in the hierarchical tree

For example, the Community called *Directors* **ImpliesAuthorityOver** the Community' called *Projects\_Leads*. The Action *Read\_All* **ImpliesAuthorityOver** *Read\_File*. The Resource *Servers* **ImpliesAuthorityOver** *Database\_Servers*.

## 2 The TCD Statutes Case Study

For the last 3 years the Trinity College Dublin (TCD) Statutes Review Working Party has been commissioned with the task of reviewing and rewriting the Trinity College Dublin Statutes. The main task for the Party is not to introduce new policies but to update and rewrite the existing ones. This well documented, real-world problem of statutes definition and interpretation in order to minimize the number of conflicts poses a big challenge. We decided to study the combination of CBPM and OWL DL techniques in order to tackle the encoding of existing statutes and the detection of conflicts when enforcing new ones. Some of the challenges this effort has, is to capture the complexity of the organizational structure of the College, with the policies and obligations pertaining to each College body.

For this purpose the use of the CBPM was chosen as a candidate to model policy subjects. Also, due to the reasoning power of OWL DL, static conflict analysis could be performed and this was one driving factor for using ontologies.

However, the main challenge that we faced was finding the right balance between the use of CBPM and OWL DL in terms of system expressivity without incurring a big penalty on the user cognitive load. Since it is envisaged that such a system would be used not only by experts in policy management, but also by domain experts (i.e. experts in law specification in this case), the creation of easy to use, custom tools that support these encodings is considered of great importance.

Such a tool has already been developed and usability studies are being performed. In the remainder of this paper however, we will not focus on the tool support, but on the task of finding a working balance between the various encoding options, during the encoding of the existing statutes. For that purpose, a user trial has been conducted which did not involve the use of any particular tool. This was done with the intention of not skewing the results with potential usability issues the tool has.

Five carefully selected participants took part in the user trial, each with a strong background on either policy-based management, ontologies, or logic programming. This is because the users would be asked to encode statutes using such techniques. The user trial involved a training session introducing all encoding methods, where examples of already encoded statutes were presented, and the actual user statutes encoding session. The whole trial was recorded and in the end the users were asked to fill in a questionnaire.

The users were presented with an “encoding path” which they should follow, in the form of a step-by-step encoding methodology. This was done so that all users irrespective of their background, will consider all encoding options and not jump straight to the one they feel more comfortable with. It was also done so that the users won’t be intimidated by the number of encoding options available to them, especially during their first encodings.

The encoding methodology given to the users was the following:



**Fig. 1.** 5-step encoding methodology

All users started with CBPM, and if it was not possible to encode with CBPM or they were not happy with the outcome, they moved to the next available encoding method, which is CBPM+DL in this case. The same happened until the users traversed all remaining encoding methods, namely OWL DL, OWL DL + LP, and LP.

During the training sessions, the users were informed that there is no single “golden” encoding, and as long as they adhered to the encoding guidelines, they were free to come up with an encoding that could differ with the one suggested by the policy expert.

The OWL DL + LP method was in essence the use of the SWRL language [10]. When selecting LP in this trial, the users had to use the syntax of a generic rule language such as Drools [11].

## 2.1 Example Statutes Encodings

### 1st Example

The following example was presented to the users during the training session. The task is to encode the following clause from the TCD statutes:

“The Chancellor shall be a member of the Caput of the Senate and shall preside at its meetings”

Following the encoding methodology, the user has to try to use CBPM first.

The user can search for nouns, which are subjects and as a result can be candidates for being CBPM communities. Likewise, the user can search for targets. For instance, this clause can be broken down as follows:

C                      NCC                      C  
 The Chancellor shall be a member of the Caput of Senate  
A                              R  
 and shall preside at its meetings.

C:Community, A:Action, R:Resource, NCC: Native CBPM Construct

The “Chancellor” is the subject and therefore it is a Community. We don’t know whether the “Chancellor” has authority over the “Caput of the Senate”, so the latter cannot be encoded as a Resource. So “Caput of the Senate” is a Community. “Preside” is an Action and “meetings” is a Resource, probably of type “Caput of the Senate” since the clause describes the “Caput of the Senate meetings”. The term “Member of” found in this clause, is an indicator that the “Chancellor” can be a member of the “Caput of the Senate” Community, and therefore can be regarded as a Native CBPM Construct.

We are now ready to encode the first part of the clause, namely “The Chancellor shall be a member of the Caput of Senate”. We will use the CBPM membership rule to achieve this:

Person(?P) ∧ memberOf(?P, Chancellor) → memberOf(?P, CaputOfSenate)

This clause can be read as follows: For Person P and P is a member of the Chancellor Community, it implies that P is a member of the CaputOfSenate Community.

We can now proceed to encode the second part of the clause, “(The Chancellor) shall preside at its meetings”. We can use a CBPM positive authorization rule:

First we define the resource authority, which does not mention anything about the subject:

Resource Authority:

Action(Preside) ∧ Resource(meetings) → ResourceAuthority(presides\_meetings)

The rule that uses the given resource authority is the following:

comm(Chancellor) ∧ ResourceAuthority(presides\_meetings) → cl.posAuth(Chancellor, presides\_meetings)

This rule can be read as follows:

For the Chancellor Community and for the presides\_meetings Resource Authority, it implies that the Chancellor is given positive authorization to preside in Senate’s meetings.

The encoding of this clause is a good fit for CBPM, because both the membership rule and the authorization rule can be applied easily. The same clause can also be encoded with the other remaining methods in the encoding methodology. For instance, for the OWL DL encoding, the clause breakdown is shown below:

OC                      OP                      OC

The **Chancellor** shall be a **member of** the **Caput of Senate**

OP                      OC

and shall **preside** at its **meetings**.

OC:Ontological Class, OP:Ontological Property

Following the OWL DL modelling, “Chancellor”, “Caput of Senate” and “meetings” can be ontological classes. “Member of” and “preside” can be ontological object properties.

Thus, so far we have encoded the same TCD statute clause using two encoding methods, CBPM and OWL DL. The encoding choice is simply a matter of personal preference. However, in some type of clauses, such as those that have conditionals, the encoding process is not obvious, as user trials have shown.

## 2nd Example

This clause includes a conditional:

“The Chancellor shall act as head of the University on ceremonial occasions”

“On ceremonial occasions” is the conditional.

Following a similar breakdown as in the previous clauses, the “Chancellor” and the “University” can be encoded as Communities and “head of” can be an Action. The question that arises is how to encode the “ceremonial occasions”.

Let’s assume that the condition “Ceremonial occasions” is encoded as a Community. This means that a new Community called Ceremonial is created and will be utilized on Ceremonial occasions.

We can use the CBPM authorization rule to encode this clause. First we create the Resource Authority:

**Action(heads)  $\wedge$  Resource(University)  $\rightarrow$  ResourceAuthority(heads\_university)**

The authorisation rule reads:

**comm(Chancellor)  $\wedge$  memberOf(Chancellor, Ceremonial)  $\wedge$  ResourceAuthority(heads\_university)  $\rightarrow$  cl.posAuth(Chancellor, heads\_university)**

The meaning of this rule is that the Chancellor, who belongs to the Chancellor Community and is a member of the Ceremonial Community and for the given Resource Authority, it implies that the Chancellor is authorized to head the University.

The encoding of this rule has shown that by restricting membership into a new Community, such as the “Ceremonial” in this case, statute conditionals can be represented.

However, during the presentation of the same example in the training session, all users agreed that the “Ceremonial” Community is an odd concept, and did not feel natural. Therefore, they were urged to seek alternative methods to encode this clause.

Using CBPM+DL, this clause would be encoded as follows:

**cl.comm(Chancellor)  $\wedge$  isa(Ceremonial, Event)  $\rightarrow$  cl.posAuth(Chancellor, heads\_university)**

Using such an encoding there is no need to create the “Ceremonial” community. Instead, using the “is-a” relationship, a more “user-friendly” Ceremonial Class is created, and we define the Ceremonial Class as a subclass of the Class Event.

This CBPM+DL encoding has thus proven to be an efficient encoding method for cases like this.

It is worth noting that using OWL DL to encode and infer conditionals is very hard, and all users avoided that.

### 3rd Example

Although OWL DL cannot encode complex conditionals, there is one type of statute clause, where it was particularly useful.

“There shall be up to six Pro-Chancellors, who shall be members of the Senate ex officio”

This clause is hard to encode in CBPM. It would require the introduction of a meta-rule (a rule about a rule) that would count the number of existing Pro-Chancellors. The CBPM syntax which is available for this user trial, does not cater for the use of meta-rules.

On the other hand, this is a clause where the OWL max cardinality restriction is ideally suited for its encoding:

Pro-chancellors is max 6

### 4th Example

The users had to encode the following clause:

“The Chancellor shall be the primary Visitor of the College and University, and in the event of a disagreement between him and the other Visitor the opinion of the Chancellor shall prevail”

One way of encoding this rule is to use a similar analysis as in the previous examples. However, a more holistic view of this clause reveals that it would be a good fit for the CBPM implies authority relationship.

In fact some users selected this method as their preferred one:

Chancellor ImpliesAuthorityOver Other Visitor.

The users have used the “implies authority” relationship to describe the relationship between the “Chancellor” and the “Other Visitor”. By doing so, they have ensured that under all circumstances, such as a disagreement between the “Chancellor” and the “Other Visitor”, the “Chancellor” always prevails. This encoding has saved the users from encoding every single predicate of the clause. For instance, the term disagreement is not encoded.

This encoding has proven than CBPM when used in such scenarios, can be very beneficial both in saving rule editing, as well as resulting in smaller encodings, which can be more easily understood by a third party.

The use of the “implies authority” relationship in this case, is an elegant solution of avoiding the painstaking task of having to analyse complex clauses.

## 2.2 User Trials

For this research, to evaluate the user cognitive load caused by policy engineering tasks, a number of metrics are currently used:



- the user prior-knowledge of the encoding method in question.
- the time required to complete all tasks.
- an intervention analysis, which studies the type of interventions performed by the policy expert, so that the users complete all tasks successfully.

These metrics are an indirect indication of the user cognitive load, since they do not include the use of any type of biometric sensor. The metrics currently used were selected on the basis that they represent an easy and unobtrusive method of collecting data. In this paper, only part of the intervention analysis is shown.

### 2.2.1 Intervention Analysis

There were two main actors when conducting the user evaluation experiments: the users –being experts in either ontologies, CBPM, or rules- were asked to perform a series of tasks and complete a questionnaire in the end, and the policy/domain expert who conducted the experiment. The intervention analysis studies the type of interventions performed by the policy/domain expert, during the experiment, as a means to assist users. The number and type of interventions can be an indication of the user cognitive load. For this user trial, which involved the encoding of TCD college statutes using a number of different encoding methods, there were 5 types of domain expert intervention.

**I1:** The user doesn't know what to do, and the domain expert gives a hint

**I2:** The user asks for clarification about the syntax and the domain expert gives a hint

**I3:** The user asks for clarification about conceptualizing the problem and the domain expert gives a hint

**I4:** The user doesn't ask for clarification, the domain expert intervenes

**I5:** The user doesn't ask for clarification, the domain expert doesn't intervene

The intervention made by the domain expert for every user and for each task is shown in Figure 2. This graph shows that during the first task, most of the users didn't know what to do and as a result, the domain expert intervened a lot. This is in stark contrast with the 5th task, where the domain expert intervened the least and three users completed the encoding without any intervention. This has to do with the fact that by the 5<sup>th</sup> task, the users had up to a certain degree acclimatized themselves with all the different encoding methods. This highlights the importance of user training.

The 1<sup>st</sup> task was about the application of the CBPM syntax, and as a result most users needed time to familiarize themselves with its syntax.

The encoding of the 2<sup>nd</sup> task was not an obvious selection, and the users had to decide on their favourite method. This is depicted by the fact that almost all users needed assistance in order to conceptualize the problem, hence the Intervention type 3 is shown as the highest for this task.

The 3<sup>rd</sup> task was about discovering that CBPM does not support meta-rules in its given syntax, and that users can use the OWL DL cardinality relationship. This is why the intervention type 2, which is related to clarifications about syntax, is shown as the highest.

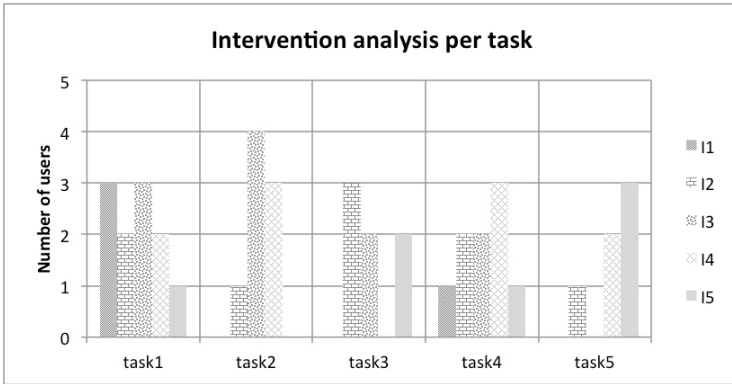


Fig. 2. Intervention analysis per task

The 4<sup>th</sup> task was about the encoding of a conditional. OWL cannot be used to encode and infer conditionals, but all other methods can be used. Most intervention types were used in this task, since it was the first time the users had to encode a conditional.

The 5<sup>th</sup> task builds on top of the 4<sup>th</sup> one, since again it describes a conditional, and that is why there are considerably less interventions needed compared to task 4.

### 3 Discussion

What has become obvious from analyzing the user responses, is that there is no one-size-fits-all encoding method, which doesn’t cause a rule clutter. Obviously, nearly every clause can be expressed in Logic Programs (LP) in the form of generic rules such as [11], but firstly, not all users are comfortable in using and interpreting LP rules and secondly the outcome being in the order of hundreds or thousands of rules, would be hard to maintain. Also, the benefit of having static rule conflict analysis that OWL DL provides, would be missing from an LP only case.

The 5-step encoding methodology used in the trial, does not necessarily describe a path from low expressivity to high expressivity, or from a given set of semantics to a richer set of semantics. This is because in some cases the semantics involved are just different, such as those between CBPM and OWL, and it is generally hard to agree on which one is semantically the richest.

What has been an interesting point of discussion is whether CBPM needs to be the first option for the encoding. This was done primarily because only after repeated use, could the users better utilize CBPM’s inherent principles. For instance, some of them used the CBPM “implies authority” relationship to describe a complex conditional.

Describing and inferring conditionals can be regarded as one of OWL DLs weakness, and thus alternative encoding methods must be selected.

Since the aim of the experiment was to showcase the strengths and weaknesses of each encoding method in a controlled environment, the syntax of each of the methods was explicitly defined and was available for the experiment participants to use. For CBPM, meta-rules were specifically omitted, since this is more akin to rule-based

systems. Therefore, an alternative encoding method needs to be selected, when meta-rules are required, e.g. the use of OWL DL as shown in the example. Also, user-defined properties were left out of the CBPM syntax, since this is more tailored to OWL DL based encodings. The introduction of CBPM+DL encoding method can be seen as a bridge between two different worlds. Using CBPM+DL in the trials, the users were able to create custom properties and intuitive encodings in the cases where CBPM produced counter-intuitive ones, according to user responses. More on the user's preferences regarding the various encoding methods can be found in [8].

## 4 Conclusions and Future Work

This research aims to assist policy experts and domain experts in the task of policy engineering, and especially in cases where the complexity of the subjects, targets and actions used in rules is high. The modelling of the Trinity College Dublin statutes poses such a challenge and for that purpose the use of semantically enhanced techniques has been suggested as a viable solution. This modelling involves the creation of a new encoding method called CBPM+DL, which uses CBPM to model organizational subjects and model the flow of authority between different parties. CBPM+DL also suggests the use of OWL DL to model changing policy resources (targets) mainly due to OWL's ability to offer a taxonomical representation of complex resource relationships as well as static policy conflicts checking. For example, OWL DL can express equivalence, and when contradicting policies are enforced onto a class, its equivalent classes can also be signalled as 'in conflict'. This can facilitate conflict detection, since there is no need for the policy author to monitor for conflicts in all equivalent classes.

This paper has shown that the use of CBPM can be beneficial in terms of increasing the overall system expressivity, but unless systematic user training is performed, it is difficult to use in practice, due to increased user cognitive load. For this user trial, the users were informed that there is no 'golden' encoding method, and as long as the outcome is the one intended, any method can be used. Potentially, this can be an issue in collaborative environments, where more than one policy authors is involved and maintaining consistency when encoding similar rules is a requirement.

Currently further trials are being performed, where the user's prior knowledge in relevant technologies is correlated to the user's performance in policy engineering tasks, such as policy conflict detection and resolution. Further evaluations are made to test the efficiency of a custom-built policy engineering tool, capable of modelling in the CBPM+DL form. This tool is based on the Graphical Modelling Framework [12], and its efficiency is compared against Drools-based graphical editors [11].

Related work can be found in [13] where Rei, a policy framework that permits the specification, analysis and reasoning about declarative policies is presented. Rei adopts OWL Lite to specify policies and can reason over any domain knowledge expressed in either RDF or OWL. KAoS [14] defines basic ontologies for actions, actors, groups, places, resources and policies. Using an inference engine, KAoS can reason about policy disclosure, conflict detection, and domain structure. SWRL [10] combines OWL with Horn-like rules. What differentiates the research presented in this paper from these approaches, is the use of Community modelling (CBPM), to

model policy subjects, and their interactions, such as the delegation of authority between Communities. CBPM+DL also emphasizes in producing encodings, which are –in some cases- more intuitive than plain CBPM, as user trials have shown, thus assisting in the reduction of the user cognitive load. Since RuleML [15] is based on an XML mark-up language, it is tailored mostly for web use, rather than for the business level. Finally, SBVR [16] can be used in the future to model CBPM+DL concepts, such as Communities, the authority transfer between Communities, as well as the formation of Community Federations. In CBPM, parts of an existing or of different organisations can form a Federation, which is delegated authority over resources.

**Acknowledgments.** The authors would like to acknowledge the Centre for Next Generation Localisation (CNGL) ([www.cngl.ie](http://www.cngl.ie)), for its support.

## References

1. Feeney, K., Tsarouchis, C., Lewis, D.: Policies as Signals in Collaborative Policy Engineering. In: Policy-based Autonomic Computing, PBAC (2007)
2. Batra, D.: Cognitive complexity in data modeling: causes and recommendations. *Requirements Engineering* 12, 231–244 (2007)
3. Antoniou, G., et al.: Combining Rules and Ontologies. A survey (2005)
4. Feeney, K., et al.: Relationship-Driven Policy Engineering for Autonomic Organisations. In: 6th IEEE International Workshop on Policies for Distributed Systems, POLICY 2005 (2005)
5. Baader, F., et al.: *The Description Logic Handbook*. C.U. Press (2002)
6. Baral, C., Gelfond, M.: Logic programming and knowledge representation. *Journal of Logic Programming* 19/20, 73–148 (1994)ss
7. OWL-overview, <http://www.w3.org/TR/owl-features/>
8. Tsarouchis, C., O'Sullivan, D., Lewis, D.: A study in the expressiveness of semantically different policy modelling schemes. In: 6th IFIP/IEEE International Workshop on Business-driven IT Management BDIM, Dublin, Ireland (2011)
9. Brennan, R., et al.: Policy-based integration of multiprovider digital home services. *Netwrk. Mag. of Global Internetwkg* 23(6), 50–56 (2009)
10. SWRL, <http://www.w3.org/Submission/SWRL/>
11. Drools, <http://www.jboss.org/drools/>
12. GMF, <http://www.eclipse.org/gmf/>
13. Kagal, L.: *Rei: A Policy Language for the Me-Centric Project*. HP Labs (2002)
14. Uszok, A., et al.: KAoS Policy and Domain Services: Toward a Description-Logic Approach to Policy Representation, Deconflictions, and Enforcement. In: IEEE 4th International Workshop on Policies for Distributed Systems and Networks (2003)
15. RuleML, <http://www.ruleml.org/~Scope>
16. SBVR. *Semantics of Business Vocabulary and Rules*, <http://www.omg.org/spec/SBVR/>