

On Harnessing Information Models and Ontologies for Policy Conflict Analysis

Steven Davy, Brendan Jennings, John Strassner
Waterford Institute of Technology,
Cork Road, Waterford, Ireland

Abstract—We present a policy conflict analysis process that makes use of pre-defined semantic models of an application to perform effective and efficient conflict analysis. The process is effective as it can be used to analyse for policy conflicts that may occur in different applications due to the separation of application specific information and constraints from the algorithms to semantic models, such as information model and ontologies. The process is efficient as it incorporates a pre-analysis policy selection step that reduces the number of policies that need to be analysed more extensively. Experimental results show that this process results in a significant reduction in the number of policies that needed to be analysed for potential conflict and that it is flexible enough to detect for policy conflict both for many popular applications and between different applications.

Index Terms—Policy-based management, Policy conflict analysis process

I. INTRODUCTION

Policies are meant to govern the behaviour of communications networks [1]. Unfortunately, most policy-based management systems do not contain policy conflict analysis processes, which makes it difficult to guarantee that the intended behaviour will be realised by the communications network. In addition, most policy-based management systems are designed to be used by a single constituency of users, so do not address the needs of different stakeholders that need to define policies for different applications. When there are multiple needs and objectives associated to an organisation, then there will also be a requirement to define policies to describe the intended behaviour of the managed entities related to each objective. These objectives may be initially defined as business objectives and later as network performance objectives. Therefore, the policies defined to meet the business objectives are inherently associated to the policies defined to meet the network performance objectives. Policy conflict can manifest differently for these different types of policies.

An important challenge, that has not been adequately addressed until now, is how to perform effective and efficient policy conflict analysis when faced with different but related levels of policies that are defined to meet the objectives of multiple stakeholders of an organisation. This paper describes the policy conflict analysis process outlined in the thesis [2], that addresses the challenges associated to analysing policies defined for multiple applications. The conflict analysis algorithm is part of a policy authoring process defined for the policy continuum [3], [4]. The policy authoring process makes the assumption that there are always multiple levels of

policies associated to each other. Once a policy (the candidate policy) is altered, it must be verified to ensure the policy continuum is still a valid representation of the organization's objectives, it must be analysed for policy conflict to ensure that the communications network behaves as intended; finally, the policy must be refined so that the communications network is configured to meet objectives of the updated policy.

The challenges faced by a policy conflict analysis process that must be compatible with the view defined above in that it must be flexible enough to discover potential conflicts for multiple applications. One of the strong agreements within the domain of policy conflict analysis is that policy conflicts can manifest differently for different applications; essentially, policy conflict analysis processes need to be made aware of these "application specific" conflicts, if it is to discover them. Until now, no policy conflict analysis process has been designed to be usable for multiple applications and be flexible enough to detect potential policy conflicts successfully for those applications.

Initially, policy conflict analysis algorithms were designed to only take into account the syntax of the policy language in use and needed to make assumptions about the application, for conflicts to be detectable. This is exemplified by the work of Al-Shaer and Hamed [5], where the application of policies was a network traffic firewall. They examine each policy rule in the firewall in a specific order and can infer if anomalies exist among the policy specifications. The nature of the approach has limitations; their algorithm does not take into account deployments of policies for other applications, such as routing and quality of service classification. Also, their algorithm is specifically tailored for use in analysing firewalls and cannot be used to detect for conflict within other applications types. This is shown in the fact that the algorithms was extended to investigate conflicts between physically distributed firewall [6].

In designing conflict analysis algorithms that are inherently more flexible, Bandara et al. [7], [8] investigated the use of external models to replace the inherent assumptions of the applications. They developed more generic conflict analysis algorithms that made use of a model based on Event Calculus. This enabled an applications expert to design the constraints of the application separately from the analysis algorithms. A related approach was proposed by Chomicki et al. [9] where they used logic programming to define conditions that may lead to policy conflicts, separately from the analysis algorithms. The drawbacks of these approaches are that they

make use of non-standard models that may be difficult to extend and enhance for new applications.

Many new policy languages such as Rei [10] and KAoS [11] use an ontology representation language, namely OWL, to define policies. They make use of the inherent reasoning capabilities of ontology reasoners to analyse for potential policy conflicts. However, they do not go far enough, for instance, Rei and KAoS can only be used to detect for application independent policy conflicts or those conflicts that can arise independent of the type of application using policies. New research that makes use of the search and query capabilities of ontologies to detect for policy conflict was presented by Kaviani et al. [12], [13]. Verlaenen et al. [14] demonstrates the use of rules and ontologies for policy analysis, and shows that very rich relationships between policies can be readily ascertained.

The work of the thesis produced a policy conflict analysis approach that makes extensive use of information models and ontologies to make it flexible enough to be used as a tool to analyse for conflict in a range of applications. It introduced a novel pre-analysis policy selection step to significantly reduce the number of more thorough policy analysis operations required to determine if a group of deployed policies are free from conflict. It uses heuristics and historical information from previous comparisons to aid in the elimination of groups of policies from analysis. It is designed to be used in policy repositories that are large or distributed.

The paper is outlined as follows, section II describes a motivating example that shows the difficulties faced with performing policy conflict analysis. Next, section III presents an overview of the main algorithms and processes presented in the thesis, and illustrates their use with respect to the example of section II. Section IV presents the policy authoring process for the policy continuum and shows the important role policy conflict analysis plays. Finally, section V presents the most interesting conclusions of the work and hints for the future of this research.

II. MOTIVATING EXAMPLE

The example is focused on showing the lack of an integrated approach to policy conflict analysis, where large numbers of policies need to be analysed. The main contributing factor to the difficulties encountered in analysing these policies, is that there may be multiple applications that use policy based management; however, there is no generic approach to analyse each application for conflicts that occur specific to that application.

The example involves multiple routing devices, capable of enforcing firewall policies and IPsec VPN policies as IP packets flow through their interfaces. There are conflicts that can occur between policies for a single application, but there are also conflicts that can occur between the policies of two or more applications. These types of policy conflicts are not detectable without the algorithms and processes being enhanced with information pertaining to the interaction between the applications. As presented in [6] and [15], there is a close relationship between the types of policy conflicts that can

occur for IPsec policies and firewall policies. However, detecting such policy conflict is exacerbated when the both firewall policies and IPsec policies are deployed simultaneously and are distributed across multiple routing devices in a network.

A typical conflict that may arise given this above example is, an intermediate router (IR) has IPsec policies installed to secure IP traffic from a source network (A) and encapsulates it into a tunnel; then an edge router (ER) has policies installed to block IP traffic originating from source network (A). One conflict that manifests is that the IPsec policies on IR disguise the source IP address of the traffic from A, and so ER's policies cannot be applied. This is an example of a distributed policy conflict involving multiple applications. Moreover, there may be business objectives defined that lead to these policies being deployed, in which case, the business policies are in conflict too. The challenges encountered here are as follows:

- How can we know which policies located across the network are related to each other, in order to retrieve those policies for policy conflict analysis?
- How can we compare policies defined for different applications together to ascertain whether a policy conflict can potentially occur?
- There may be potentially thousands of policies defined for routers in a communications network, how can we reduce the number of comparisons we need to perform in this case?

III. EFFICIENT POLICY CONFLICT ANALYSIS

The policy conflict analysis approach presented in the thesis comprises of three main steps. The first step is the pre-analysis policy selection step that makes use of ontologies and rules to quickly reduce the number of policies required for analysis. Ontologies can be used to represent semantically rich relationships between concepts, and can thus be used to accurately identify concepts related to policies. The second step is the use of tree data structures to eliminate the number of comparisons required for the conflict analysis, this is built by re-using the results of previous policy comparisons. The third step is the policy conflict analysis algorithm, here a policy relationship matrix is built between a candidate policy (new or modified) and a selected deployed policy. This policy relationship matrix is then analysed against a pre-defined policy conflict matrix that can investigate if a set of required relationships between the two policies holds true.

The approach is extensible because the definition of a policy conflict is separated from the definition of the conflict analysis algorithm. This separation comes from the fact that the conflict matrix can be designed for new applications. Application specific information is encompassed in information models and associated ontologies to enable the modeling of rich application semantics that would ordinarily not be available when using UML (for information models) alone. The conflict matrix is essentially incorporated into the information models; therefore, by extending the information model, new conflicts types can be identified.

The approach is efficient in two ways, the use of ontology based rules, termed selection rules, are an effective method

of investigating if groups of policies should be considered for analysis. Also, the use of structured trees used to store policies promotes the use of tree traversing algorithms in reusing the results of previous comparisons, instead of re-computing values which may be a time consuming process.

A. Pre-Analysis Policy Selection

Selection rules defined using ontologies are inherently extensible as ontologies are designed to be extensible. These rules are used to perform a lightweight analysis step across all policies, to essentially flag those policies that meet the criteria set out by the selection rules. With regards to the example mentioned in section II, the selection rules would be designed to ensure that from all the policies that can be considered for conflict analysis (all policies distributed across the communications network), only those that are related to firewall and IPsec are selected. Moreover, local firewall policies were selected first, then only those policies located on routers en-route to the destination network as defined by the destination address of the candidate policy. The selection rules can also be defined to retrieve IPsec firewall rules. The selection rules are future proof as the ontology can be readily extended so that new types of policies can still be retrieved given that they can be classified under existing concept types. More details on selection rules can be found in [16].

B. Tree-based Policy Selection

The method of establishing relationships between two policies entails performing a set of operations between the individual components of the policies. The components of policies assumed in the thesis are events, conditions, actions, subjects and targets. By representing policies as a tuple of sets, then it is possible to build sets of functions to evaluate how tuples can be related to each other. As a candidate policy is compared against a deployed policy, the result is a policy relationship matrix. However, if two deployed policies share a number of components in common, then for that shared set of components, policies comparisons can be stored and reused. This is the essential concept behind the policy trees. Policies that share common components can be located at the same node in the tree. There is a different tree per policy component type; therefore, we assume there are five trees that can be used to relate policies together.

Given the example mentioned in section II, we can define a policy tree to represent the relationships between policy conditions, where the condition of the policy relates to the IP header field match criteria. Policies with the same match criteria can be located in the same node of the tree. Policies that cover a super set or subset of another policy's match criteria can be used to create hierarchical nodes in the tree. Note that IPsec policies and firewall policies can be related together in these trees as they use the same method of identifying IP headers. The benefit of this approach becomes apparent in the next step of the process, where relationships between a candidate policy and currently deployed policies can be readily identified without repeating the evaluations of policy components. As presented in [17] (see figure 1), experiments

demonstrated a reduction in comparisons of up to 60% for a distributed firewall analysis scenario.

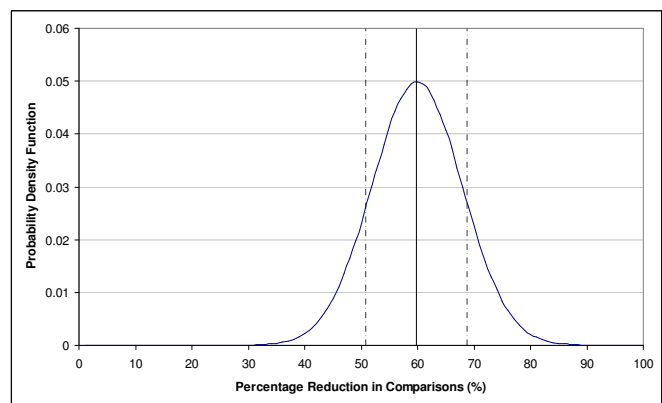


Figure 1. Percentage reduction in comparisons

C. Conflict Analysis Algorithm

The previous two steps of the approach are concerned with reducing the number of policies and comparisons required to test a group of deployed policies for policy conflict with respect to a candidate policy. This step is concerned with actually indicating, on a pair-wise basis, whether two policies potentially conflict or not. The method that is presented in the [2], and in more detail in [18], [19], [4], makes use of an information model based policy relationship matrix. The policy relationship matrix is a set of entries, where each entry relates to a potential relationship that can be ascertained between two policies. There is a row in the matrix for each policy component type, for our usage, there are five rows in the matrix. Note that depending on the policy model in use, more or less rows may be defined. The process for creating a new row for a new policy component type is outlined in the thesis. Each row contains a number of relationship entries; each entry represents a particular relationship. For example, there may be an entry in the first row relating to subject equality. This means that the first row of the matrix is concerned with enumerating relationship related to the subjects of a policy, and that one of these relationships refers to an equality evaluation function. If this function evaluates to true, then a 1 is placed in the matrix in the appropriate position. Figure 2 shows the template of the matrix. The letters of each row in the matrix shows the relationships that are considered initially. The first letter indicates the policy component type, *s* for subject, *t* for target, *e* for event, *c* for conditions and *a* for actions. The remaining letter describe the nature of the matrix entry, *sb* for subset, *sp* for super set, *eq* for equal, *cor* for correlation, *mux* for mutually exclusive and *ctd* for contradiction.

The matrix is populated by evaluating each entry of each row for a pair of policies; this describes the relationship between two policies. It is these relationships that must be further examined if a potential conflict is to occur. However, depending on the type of applications that the policies are being deployed for, the relationship matrix may not have enough

$$\begin{bmatrix} ssb & ssp & seq & scor & 0 \\ tsb & tsp & teq & tcor & 0 \\ esb & esp & eeq & ecor & emux \\ csb & csp & ceq & ccor & cmux \\ asb & asp & aeq & acor & actd \end{bmatrix}$$

Figure 2. A policy relationship matrix

information to actually indicate the potential occurrence of a policy conflict. Therefore, a policy conflict matrix is defined on a per application basis. The policy conflict matrix is designed by a policy specification expert for a particular application type. The policy conflict matrix indicates those relationships that should exist if a potential policy conflict should occur between two policies.

The types of relationships that can be tested for, and subsequently used to detect potential policy conflict, are designed in the information model as subclasses of the class PolicyRelationship. The DEN-ng information model [1] is used in the thesis as a base model to validate the approach. By separating the details of the two types of matrices from the algorithm and into the information model, then the form that each takes may be different depending on the types of policies being used and the types of policy conflicts that need to be analysed for. In the example mentioned above, the types of relationships that are of most importance are the contradicting actions relationships. These relationships are designed to indicate that the tunnel encryption action may contradict with the drop actions of firewalls. Equation (1) depicts the evaluation operation between the policy relationship matrix and the policy conflict matrix. The use of the evaluation may be altered depending on the re-use of previously evaluated functions. The evaluation is independent of the dimensions and size of the matrices defined previously.

$$(M \otimes M)_{i,j} \rightarrow B$$

$$(a \otimes b)_{i,j} \triangleq \bigwedge_{p=0}^i \bigvee_{q=0}^j (a_{p,q} \wedge b_{p,q}) \quad (1)$$

Semantically rich relationships can also be incorporated into the policy relationship matrix to indicate that some policy components are related via an ontological relationship. It was shown in [16] that new relationship types based on information from ontologies can be used to augment the relationship matrix. This lead to a more expressive policy relationship matrix, where relationships based on classification, restrictions and logical inference can be leveraged to aid in describe and analysing for policy conflict.

The novel aspect of the approach is that it can be extended to suit different applications and takes into account application specific information in the form of information models and ontologies to describe what constitutes a policy conflict for that application. An added benefit of this approach is that it can be used as part of an over arching policy authoring process that requires an approach to policy conflict analysis that can be reused to for many different types of applications.

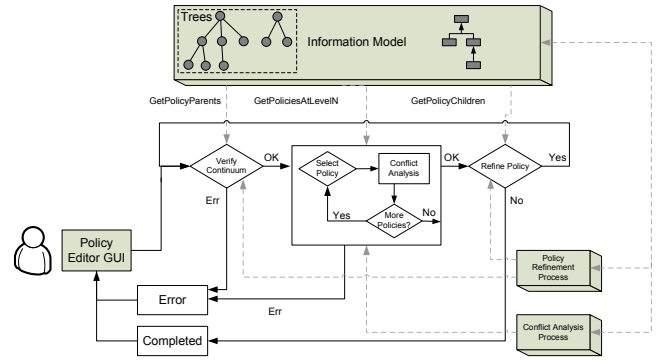


Figure 3. Policy authoring process, including policy selection and conflict analysis.

IV. POLICY AUTHORIZING PROCESS

The policy continuum is not formally described by Strassner [1], and so it is not clear how policy authoring and policy conflict analysis should operate in this context. This section describes how the policy authoring process can make use of the policy conflict analysis approach defined above. The role of the policy authoring process is to define the steps that a policy author must go through in order to have a candidate policy incorporated into the policy continuum, ensuring that the policy continuum remains valid. There are a number of reasons why the policy continuum may become invalidated due to the authoring of policies:

- 1) The candidate policy may no longer satisfy the objectives of policies defined at a higher level of the policy continuum.
- 2) The candidate policy may contradict with policies defined for applications deployed at the same levels.
- 3) The candidate policy may lead to the creation / modification of policies at lower levels that may invalidate the policy continuum according to the previous two points.

For these reasons, a policy authoring process is required. A formal version of the policy continuum is presented in [3], [4], where the relationships between policies defined at the multiple levels of the policy continuum can be queried and retrieved to aid in policy authoring. The proposed policy continuum model is based on a hierarchical sets of overlapping trees of policies, where the root node of each tree is a policy specified at the highest level (in this case the business level). We designed tree traversal algorithms to identify overlapping trees and hence indirectly related business policies. The policy authoring process makes use of these tree traversal algorithms to retrieve sets of policies for input into the algorithms developed for analysis within the thesis.

The authoring process is outlined as follows and shown in figure 3:

- 1) **Create/Modify Policy:** The policy author creates or modifies a policy that is added to the policy continuum (the candidate policy).
- 2) **Verification:** The candidate policy is used by a policy verification process to investigate if the goals of higher-level policies are affected by the inclusion of the candi-

date policy, if not the process continues. Note that the verification process is mentioned but not defined in the thesis.

- 3) **Conflict Analysis:** The candidate policy now has to be analysed for conflict between the policies at the same policy continuum level.
 - a) **Select Policies:** The selection algorithm makes use of the policy type (class) information to retrieve the appropriate selection rules represented in the ontology. These rules are then used to select a subset of deployed policies that should be further analysed for policy conflict with the candidate policy. This step is performed before the policy trees are considered in order to restrict the number of policies that need to be considered for conflict analysis, and to leverage semantic information available in the ontology.
 - b) **Select Policy Trees:** The selected policies have pre-existing relationship information stored in the policy trees that relates those policies to other deployed policies. The policy trees are retrieved based on the type of relationships that can be established in the policy relationship matrix. Essentially, there is a policy tree for each row of the policy relationship matrix.
 - c) **Add Policy to Trees:** Before the conflict matrix is considered, the candidate policy is efficiently added to each tree, per relationship type. This step effectively evaluates how the candidate policy is related to all other (selected) deployed policies with a reduced number of policy comparisons. After the policy has been added to all policy trees, it has established relationships between the policies selected from step (a). The conflict matrix is retrieved from the information model and is used to investigate the nature of the relationships just established.
 - d) **Considering the Conflict Matrix:** The conflict matrix dictates the relationships that should exist for a potential conflict to occur. Each row of the conflict matrix is associated with a particular policy component type. For each row of the conflict matrix, the associated policy tree is considered. The policy trees are **reduced** to contain only those policies selected from step (a). If a "1" is in the conflict matrix, then the associated policy tree is queried to enumerate all deployed policies that are connected to the candidate policy with respect to the highlighted relationship. For example, if a "1" existed in the subject row indicating that equality contributed to conflict, then all equal policies are retrieved from the subject component tree. All other trees are successively **reduced** to contain only selected policies.
 - e) **Iterate for Each Tree:** The process is **repeated** for each row of the conflict matrix. The policy trees are searched to retrieve sets of policies that

are associated with the input policy via a specific relationship. The process stops if all conflict matrix rows have been explored, or if no policies remain selected as a result of a reduction step. If there are policies remaining after each row of the conflict matrix is considered, then these policies may potentially conflict with the candidate policy. The candidate policy then should be removed from all affected policy trees. If there are no policies remaining, then there are no policies that can conflict with the candidate policy.

- 4) **Refinement:** If there are no potential policy conflicts detected, then the candidate policy is refined by the policy refinement process. For each new policy at the lower level of the policy continuum, the process is repeated. Note that the refinement process is mentioned but not defined in the thesis.

With respect to the example mentioned in section II, the authoring process is designed to analyse the modification of business level policies that may be refined to firewall policies, and business policies that may be refined to IPsec policies. If a conflict is manifested between the business policies, then it may be detected and thus avoid the deployment of the firewall policies and IPsec policies. However, if the conflict is not detected at the business level for some reason, then a new conflict may be detected at the lower, network level. This new conflict will be detectable between the IPsec policies and firewall policies directly. One of the key advantages of this is that by detecting policy conflicts at a higher level, less policies may need to actually analysed to detect a potential policy conflict. For more information concerning the tools and testbeds used to evaluate this work see [20].

V. CONCLUSIONS AND CONTRIBUTIONS

This paper provided a summary of the work presented in the Ph.D. thesis [2] and associated publications [21], [22], [18], [3], [19], [17], [4], [16]. This paper presents an approach to policy conflict analysis that makes extensive use of semantic models to separate the definition of policy conflicts from the use of policies. The approach assumes that there exists sufficient information within these semantic models to represent the rules and constraints associated to the particular application for which policies are being defined. This assumption was supported by case studies in [2] that illustrate the information expected within the information model for a set of popular applications, including access control of distributed systems and communications network configuration. The raising exploration of using information models and ontologies for use in network management has increased the popularity and availability of tool support. It is anticipated that approaches such as the one presented in this paper will become more common place as self-knowledge becomes paramount for improving the effectiveness and reliability of management systems.

One such area of research that is investigating the use of semantic models is autonomic network management. The objective of autonomic network management is to embed self-management capabilities into the communications network.

The Framework Programme 7 research project entitled Autonomous Internet (AutoI) is exploring the use of semantic models and the policy continuum to develop a policy based virtual resource overlay for the management of future Internet architectures. The work carried out in [2],[23] and [24] is being extended in AutoI to provide a customised policy analysis process and accompanying policy continuum for use particularly for autonomous communications.

For large communications networks, there may be a lot of policies deployed, thus the overhead associated to policy analysis needs to be minimised. This paper describes the work carried out in reducing this overhead, specifically the introduction of the pre-analysis policy selection step. Selection rules are used to guide the selection of appropriate policy rules and can be defined per policy type and per application. The thesis demonstrated the advantage of using these selection rules for the analysis of distributed firewall policy analysis. Another step to improve the efficiency of policy conflict analysis was achieved through the use of tree data structures to store the results of policy relationships between previously deployed policies that can be re-used.

The main contributions of the work are that it formalised the policy continuum, it formalised the policy authoring process, it developed a policy conflict analysis process that significantly progressed beyond the state of the art. The work also showed that the use of semantic models, such as information models and ontologies, can be effectively used to separate the application information from the detection of policy conflicts, thus yielding a flexible approach that can be tailored for a range of applications.

Future work will investigate applying the algorithms and processes developed in [2] to applications that will rely on policies that may be altered continuously. Such scenarios as autonomous communications network management, cognitive networking, and distributed storage systems. Two of the processes that this thesis relies on is that of policy verifications and policy refinement, although much work has progressed in these areas, some of the efficiencies introduced in the work presented here may be leveraged.

ACKNOWLEDGMENTS

This work was funded by Science Foundation Ireland via grant numbers 03/CE3/I405 ("Autonomic Management of Communications Networks and Services") and 08/SRC/I1403 ("Federated, Autonomic Management of End-to-End Communications Services") and the EU Framework Programme 7 under the Autonomous Internet project grant no. 216404.

REFERENCES

- [1] J. Strassner, *Policy-Based Network Management*. Morgan Kaufmann, 2003. ISBN 1-55860-859-1.
- [2] S. Davy, "Harnessing information models and ontologies for policy conflict analysis," *Ph.D. Thesis, Waterford Institute of Technology*, 2008. http://www.tssg.org/people/sdavy/2008_SDavy_Thesis.pdf [available 10/11/2008].
- [3] S. Davy, B. Jennings, and J. Strassner, "The Policy Continuum - A Formal Model," in *Proc. of the 2nd IEEE International Workshop on Modelling Autonomous Communications Environments, MACE*, pp. 65–79, October 2007.
- [4] S. Davy, B. Jennings, and J. Strassner, "The Policy Continuum - Policy Authoring and Conflict Analysis," in *Elsevier Computer Communications*, vol. (31) 11, pp. 2981–2995, 2008.
- [5] E. Al-Shaer and H. Hamed, "Firewall policy advisor for anomaly detection and rule editing," in *Proc. of the Eight IEEE/IFIP International Symposium on Integrated Network Management, IM*, pp. 17–30, 2003.
- [6] E. Al-Shaer, H. Hamed, R. Boutaba, and M. Hasan, "Conflict classification and analysis of distributed firewall policies," *IEEE Journal on Selected Areas in Communications, JSAC*, vol. 23, pp. pp 2069–2084, 2005.
- [7] A. K. Bandara, E. C. Lupu, and A. Russo, "Using Event Calculus to formalize policy specification and analysis," in *Proc. of the 4th IEEE Workshop on Policies for Distributed Systems and Networks, POLICY*, pp. 1–14, 2003.
- [8] M. Charalambides, P. Flegkas, G. Pavlou, A. Bandara, E. Lupu, A. Russo, N. Dulay, M. Sloman, and J. Rubio-Loyola, "Policy conflict analysis for quality of service management," in *Proc. of the Sixth IEEE International Workshop on Policies for Distributed Systems and Networks, POLICY*, pp. 99–108, 2005.
- [9] J. Chomicki, J. Lobo, and S. Naqvi, "Conflict Resolution Using Logic Programming," *IEEE Transactions on Knowledge and Data Engineering, TKDE*, vol. 15, pp. pp. 244–249, 2003.
- [10] L. Kagal, T. Finin, and A. Joshi, "A Policy Language for a Pervasive Computing Environment," in *Proc. of the 4th IEEE International Workshop on Policies for Distributed Systems and Networks, POLICY*, pp. 63–74, 2003.
- [11] A. Uszok, J. M. Bradshaw, R. Jeffers, N. Suri, P. Hayes, M. R. Breedy, L. Bunch, M. Johnson, S. Kulkarni, and J. Lot, "KAoS policy and domain services: Toward a description-logic approach to policy representation, deconfliction, and enforcement," in *Proc. of the 4th IEEE International Workshop on Policies for Distributed Systems and Networks, POLICY*, pp. 93–96, 2003.
- [12] N. Kaviani et al., "Exchanging Policies between Web Service Entities using Rule Languages," in *Proc. IEEE Congress in Services, SERVICES*, pp. 57–64, 2007.
- [13] N. Kaviani et al., "Web Rule Languages to Carry Policies," in *Proc. of the Eighth IEEE International Workshop on Policies for Distributed Systems and Networks, POLICY*, pp. 188–192, 2007.
- [14] K. Verlaenen et al., "Policy Analysis Using a Hybrid Semantic Reasoning Engine," in *Proc. Eight IEEE International Workshop on Policies for Distributed Systems and Networks, POLICY*, pp. 193–200, 2007.
- [15] H. Hamed and E. Al-Shaer, "Taxonomy of conflicts in network security policies," *IEEE Communications Magazine*, vol. 44, no. 3, pp. pp. 134–141, 2006.
- [16] S. Davy, B. Jennings, and J. Strassner, "Using an Information Model and Associated Ontology for Selection of Policies for Conflict Analysis," in *Proc. of the Ninth IEEE International Workshop on Policies for Distributed Systems and Networks, POLICY*, 2008.
- [17] S. Davy, B. Jennings, and J. Strassner, "Efficient Policy Conflict Analysis for Autonomic Network Management," in *Proc. 5th IEEE Workshop on Engineering of Autonomic and Autonomous Systems, EASe*, pp. 16–24, 2008.
- [18] S. Davy and B. Jennings, "Harnessing Models for Policy Conflict Analysis," in *Proc. Autonomous Infrastructure, Management and Security, AIMS*, pp. 176–180, 20–22 June 2007.
- [19] S. Davy, B. Jennings, and J. Strassner, "Application Domain Independent Policy Conflict Analysis Using Information Models," in *Proc. IEEE/IFIP Network Operations and Management Symposium, NOMS*, pp. 17–24, 2008.
- [20] K. Barrett, S. Davy, J. Strassner, B. Jennings, S. van der Meer, and W. Donnelly, "A Model Based Approach for Policy Tool Generation and Policy Analysis," in *Proc. IEEE Global Information Infrastructure Symposium, GIIS*, pp. 99–105, 2007.
- [21] S. Davy, B. Jennings, and J. Strassner, "Policy Conflict Prevention via Model-driven Policy Refinement," in *Proc. of the 17th IFIP/IEEE Distributed Systems: Operations and Management, DSOM*, pp. 209–220, 2006.
- [22] S. Davy, K. Barrett, S. Balasubramaniam, S. van der Meer, B. Jennings, and J. Strassner, "Policy-based Architecture to Enable Autonomic Communications A Position Paper," in *Proc. of Workshop in Autonomic Communication at IEEE Consumer Communications and Networking Conference, CCNC*, January 2006.
- [23] S. Balasubramaniam, D. Botvich, B. Jennings, S. Davy, W. Donnelly, and J. Strassner, "Policy-constrained bio-inspired processes for autonomous route management," to appear in *Elsevier Computer Networks, COMNET*, 2008.
- [24] J. Strassner, J. N. de Souza, S. van der Meer, S. Davy, K. Barrett, D. Raymer, and S. Samudrala, "The design of a novel context-aware policy model to support machine-based learning and reasoning," to appear in *Journal of Network and Systems Management, JNSM*, 2008.