Gavin A. Campbell. Sensor Network Policy Conflicts, in L. du Bousquet and J.-L. Richier, editors, Proc. 9th. Int. Conf. on Feature Interactions in Software and Communication Systems, IOS Press, Amsterdam, September 2007.

Sensor Network Policy Conflicts

Gavin. A. Campbell

Computing Science and Mathematics, University of Stirling, Stirling FK9 4LA, UK
gca@cs.stir.ac.uk

Abstract. Policy conflict is the equivalent of feature interaction in traditional services. Conflicts between the actions of policies occur at execution time. Potential conflicts must be detected and resolved. Using an established policy system and core language, a specialised policy language for wireless sensor network management has been defined. This short paper describes the policy language and discusses policy conflicts, external sensor network restrictions, and possible approaches to resolving these issues.

Keywords: Ontology, Policy, Policy Conflict, Sensor Network

1 Introduction

A good analogy can be found between feature interaction as understood in telephony and policy conflict as found in policy-based management. This paper describes a policy-based approach being developed for the PROSEN project (*http://www.prosen.org.uk*). The project is developing condition monitoring techniques using wireless sensor networks, with wind farms being used to validate the approach.

Within PROSEN, policies are designed to monitor and manage the sensor network. When two policies become eligible for execution at the same time, their actions may conflict. Potential conflicts must be detected and resolved appropriately. Policy conflict detection and resolution are discussed for sensor networks.

The ACCENT project developed a policy-based system for (Internet) call control [1]. The approach employs a policy language named APPEL [2] with a generic foundation that is extended for particular application domains. Using the core APPEL language as a base, a policy language for sensor network management has been defined.

Section 2 describes some commonly anticipated conflicts in sensor network policies, while section 3 suggests methods of resolving such policy conflicts.

2 Sensor Network Policy Conflicts

2.1 Policy Language Structure

APPEL is defined by a core language schema that is extended for each application domain. In particular, extensions define triggers, condition parameters and actions specific to the domain. The call control language specialisation uses a large selection of triggers and actions, each with a small set of parameters (between zero and two). The policy language for sensor network management is radically different in detail, reflecting the obvious distinction between these fields. Sensor network policies have a single external trigger or action for communicating with an external entity such as a sensor node, an operator console or software agent.

A trigger is carried by `device_in`, while an action is carried by `device_out`. These carry five parameters, all strings. The 'type' defines the nature of the trigger or action, such as 'sensor_reading' or 'display_alert'. The 'entity' identifies the external component, with the 'instance' being a unique identifier for this. The 'period' defines duration reported in a trigger or the time for which an action should be performed. Finally, 'parameters' carries string of values that qualify the trigger or action. The 'type' and 'entity' parameters are mandatory, while the others are optional. The result is a very simple but powerful language. The approach to the language design is discussed in [3], while a complete definition of the language and sample policies are detailed in [2].

As there is only one external sensor network policy action, conflicts may occur between the parameters of a pair of 'device_out' actions. Conflicts must be detected and resolved by the policy system. Situations that may result in policy conflict are described in the following subsections.

2.2 General Policy Conflicts

Conflict occurs when two policies attempt to set same entity instance simultaneously. For example, one policy may wish to set the wind speed reporting interval on sensor node 53 to 10 minutes, while another policy wishes to set this to 20 minutes. Similar conflicts could arise when setting threshold values or any entity parameter. The 'parameters' string changes format depending on the 'type' of action, allowing detection of conflicting values.

A policy action might also conflict with a prior action and not just a concurrent action. The approach therefore allows conflicts to be detected between actions and states (which incorporate a history of actions).

2.3 Goal Conflicts

Conflicts can also occur between policies and goals. A goal is a high-level operational objective defined by a human operator (e.g. 'maximise sensor battery life'). A goal is realised by refinement into a set of suitable policies in order to achieve it (e.g. 'use compression to minimise transmission time', 'transmit routine data only every hour').

Issues could arise when a policy attempts to change a value which inadvertently conflicts with a goal. For example, the goal of conserving battery power conflicts indirectly with a policy that requires wind speeds over 20 m/s to be reported immediately. In addition, conflicts may occur between goals themselves (e.g. the battery life goal conflicts with the goal of reporting significant anomalies promptly).

2.4 External Resource Conflicts

The conflicting situations discussed so far consider conflicts that arise within goals and policies. A further kind of conflict can arise within the sensor network itself. Sensor nodes have limited memory, bandwidth, electrical power and processing capacity. Such constraints affect the ability of a sensor to perform multiple actions simultaneously. For example, one policy might require a sensor node to compute a cross-correlation of received data while another policy might require all data to be checksummed using a CRC. Superficially these actions are conflict-free, but as both are processor-intensive they cannot be carried out simultaneously. While this type of conflict could be viewed as out of the scope of the policy system, it is nonetheless desirable to account for resource restrictions.

3 Policy Conflict Resolution

In the ACCENT approach, resolutions are defined by policies that share the same structure as regular policies, but differ in their triggers and actions. Resolutions may be generic or specific to a sensor network. The following subsections give examples of conflict resolution for sensor networks.

3.1 General Policy Conflict Resolutions

A generic resolution selects one of the conflicting actions based on some general attribute. For example, it may select the earlier defined policy or the one that applies to a higher domain (e.g. all wind speed sensors rather than a particular one). The APPEL language allows a policy to have a preference (i.e. priority) as to how strongly the policy should be applied in the event of conflict. Possible preferences are 'must', 'should' and 'prefer', with negative forms of these plus 'don't care'. A generic resolution might select the more strongly preferred policy. Generic resolutions are useful for similar kinds of actions.

New generic resolutions for sensor network management might include choosing the action which best aids in achievement of a goal. This would involve analysing the actions as part of the goal-refinement process to determine if one action is more effective.

Specific resolutions for policy conflict are the kinds of actions that a sensor network can perform. For example, a human operator might be alerted to the conflict and ask to take action. If the conflict is deemed to be non-critical (say, backing up logs), it might be resolved by delaying one action.

3.2 Goal Conflict Resolution

The resolution of conflicts involving goals has not yet been implemented and is largely in the early stages of consideration. Much as for policies, a goal might be defined with a preference as to how strongly it must be adhered to. Resolution may be deferred to goal refinement when operational policies are derived.

3.3 External Resource Conflict Resolution

To avoid sending actions that could overload the sensor resources, it is necessary to note external resource conflicts within the policy system. One solution might be to model resource information in the ontology that has already been developed for sensor networks. Each resource could be described along with its constraints. For example, limitations could include data (bandwidth, processing), parameterisation (memory, processing), and computation (processing, memory, power). This information could be used by the policy system to link with possible action/parameter combinations and detect situations which may cause resource conflicts.

4 Conclusion

The topic of conflicts among policies for sensor networks has been discussed, along with possible resolution methods. Conflicts occur among policy actions. For sensor network policies, there is a single external action so conflicts arise among parameter combinations. A conflict occurs when two actions attempt to perform the same task on the same entity instance. Clashes among policies and goals (high-level system objectives) or among goals are also anticipated. Conflict resolution involves selecting one of the conflicting actions (generic resolution) or executing specific actions.

Further conflicts may occur external to the policy system due to resource limitations such as memory, power, processor capacity and bandwidth. Such constraints mean two seemingly viable policy actions might not be possible concurrently due to resource limitations. New techniques to model such circumstances are required, making use of resource characteristics modelled in an ontology.

Acknowledgements

The work here on the PROSEN project was supported by the UK Engineering and Physical Sciences Research Council under grant C014804. The policy language for sensor network management was developed jointly with my supervisor Prof. Ken Turner. The policy system and core policy language were initially developed by members of the ACCENT project. Thanks are also due to the PROSEN team at the Universities of Essex, Lancaster and Strathclyde.

References

1. S. Reiff-Marganiec and K. J. Turner. The ACCENT policy server. Technical Report CSM-164, Department of Computing Science and Mathematics, University of Stirling, UK, Dec. 2005.
2. S. Reiff-Marganiec, K. J. Turner, and L. Blair. APPEL: The ACCENT project policy environment/language. Technical Report CSM-161, Department of Computing Science and Mathematics, University of Stirling, UK, Dec. 2005.
3. K. J. Turner, G. A. Campbell, and F. Wang. Policies for sensor networks and home care networks. In M. Erradi, editor, *Proc. 7th. Int. Conf. on New Technologies for Distributed Systems*, pages 273–284. Cana Print, Rabat, Morocco, June 2007.