ACCENT (EPSRC GR/R31263)

(Advanced Call Control Enhancing Network Technologies)

Profs. K. J. Turner and E. H. Magill

Computing Science and Mathematics, University of Stirling

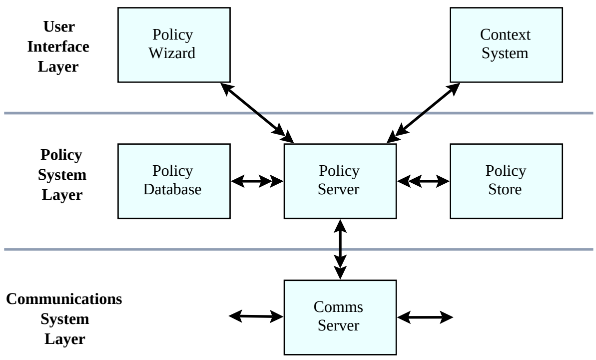
1 Background and Context

At the time of the proposal, the investigators predicted a major take-up of Internet services based on SIP (Session Initiation Protocol). This proved to be accurate, with substantial international interest developing in this field. Since the project started, many companies have announced SIP-based solutions for telephony, as well as presence and availability services. SIP was also adopted as the signalling protocol for 3G mobile communications. Most of the SIP-based developments have been commercial and pragmatic. Through the project, Stirling was able to establish distinctive expertise in control of SIP-based communications services. Few other Universities are known for this, a notable exception being Columbia (New York).

The investigators also correctly predicted a significant new demand for personalised control of communications. International developments have brought widespread use of Internet telephony (SIP, Skype), presence and availability services, instant messaging (IM, ICQ), and mobile communication (2.5G, 3G, Blackberry). Increasingly, companies have recognised the value of personalised services instead of raw bandwidth and technology.

The concept of policy-based call control did not exist when the proposal was written. The project's research on this topic remains an almost unique contribution; only work at the University of Milan has addressed similar goals. However, this has not been a purely academic exercise. The project results have been developed in conjunction with Mitel as industrial sponsor, and have been presented to a number of companies (mostly North American). The consistent feedback has been that project results are very timely, and suitable for future commercial development.

2 Key Advances and Supporting Methodology



2.1 Policy System Architecture

A three-layer architecture was defined for policy support [4, 6, 11]. The communications system layer is the lowest. This contains the communications infrastructure that is controlled by the policy system. Care was taken to make the policy system protocol-independent so that it could be used with a variety of communications systems. For example, the project approach has been demonstrated with SIP, H.323, conventional telephony, mobile telephony, PBXs and wireless devices (Blackberry, cellphone). A well-defined and protocol-neutral interface maps between call events in the communications system layer and events in the policy system layer. Communications servers and policy servers are associated in a flexible manner (1:1 or many:1).

The policy system layer is in the middle. This is where policies are stored and enforced. A policy server is responsible for retrieving policies applicable to a call. Such policies may include those of the caller and the callee, as well as higher-level policies for the domains they belong to. The policy server then detects and resolves conflicts among this pool of policies. The result is a set of compatible actions, passed in a protocol-independent manner to the communications system layer.

The user interface layer is the highest. This is where users interact directly with the policy system. A policy wizard allows users to formulate and edit policies. Considerable efforts went into making policies accessible to non-technical users. The policy wizard is web-based so that it can be used anywhere and on any system. On-line help and advice on problems are provided. The policy wizard is multi-lingual, allowing policies to be defined in near natural language. The user interface layer also provides a context system. This allows policies to be influenced by contextual information such as the user's engagements (from a digital diary), the user's role (based on organisational information), and the user's location (based on, say, an active badge system). The context system is mainly a framework for using this kind of information; a prototype was built using information from user diaries.

2.2 Policy Language

Requirements on a policy language for call control were obtained by systematically analysing a large number ($\sim$80) of use case scenarios. Call control presents a number of unique challenges that are not found in other policy-based applications. In policy-based management, there is often a well-defined structure such as clients wishing to access fixed resources according to various management policies. The situation is much more fluid and dynamic in call control. Commonly used policy concepts such as source and target do not map very well onto the elements of call control. Any subscriber may define individual policies, which have to interwork with those of any other user. In addition, multi-way calls involve the policies of several parties. This leads to a much more distributed and unpredictable policy environment, with policy conflict a very real issue.

Call control policies must be practical. For example, they require real-time detection and resolution of conflicts so that call setup is not unduly delayed. Many policy-based applications are specified by technical specialists, who can be expected to understand the nuances of policies and policy conflicts. For call control, the users are ordinary subscribers and so require non-technical ways of formulating policies and understanding their consequences. The design of the policy language was strongly influenced by the need for practical and understandable policies. Yet, the policy language also had to be capable of subtlety where required. Simple policies must be easy to express, but complex policies must also be possible where required.

The requirements for call control policies were evaluated against a wide range of existing policy languages. Although deontic logics had been considered at the time of the proposal, closer investigation raised serious doubts about their practicality and real-time behaviour. Of major policy languages, Ponder seemed the closest match to the project needs. A three-month evaluation of Ponder was conducted to check its suitability for call control policies [5]. Although Ponder was found to be partly suitable, the distinctive characteristics of call control meant that Ponder had to be bent out of shape to achieve what was needed. The style of Ponder also did not lend itself to adaptation for use by ordinary subscribers. However, the evaluation of Ponder was valuable in determining the final capabilities of the call policy language.

It was concluded that call control required a unique form of policy language to overcome limitations of existing languages. The new language, APPEL (ACCENT Project Policy Environment/Language), builds on the strengths of existing policy languages. A reference manual has been published for APPEL [9]. Practical use of the language has been described in a number of papers [1, 3, 4, 6, 11]. APPEL conforms to the commonly used ECA model (Event-Condition-Action) as this allows a spectrum of policies to be defined, ranging from high-level goals (e.g. 'employees may not make personal calls') to low-level features (e.g. 'if John is unavailable, forward my call to Mary'). APPEL has the following characteristics:

- Policies may contain modalities and user preferences. These become particularly significant when conflicts have to be resolved (e.g. 'the caller must speak to John' vs. 'the callee prefers to forward calls to Mary').
- Policies may be hierarchical. For example, policies may be defined at individual, department and organisation level (e.g. ehm@cs.stir.ac.uk, cs.stir.ac.uk, stir.ac.uk). Policy domains may also overlap (e.g. staff@cs.stir.ac.uk and research@cs.stir.ac.uk).
- Policies contain triggers, conditions and actions. These may be combined in subtle ways. For example, parallel or non-deterministic policy execution may be specified. Conflict resolution may require rearranging the order of actions or resolving non-deterministic choices.
- The policy language was carefully separated into a core ontology and application-specific ontologies. Essentially, the core language does not identify particular triggers, conditions and actions – these are defined by the application. This makes the policy language readily extensible. The project benefited from this clean split, as it allowed two extensions of APPEL to be developed: one for call control, and one for policy conflict handling. For call control purposes, policies are defined in a protocol-independent manner. This permits a

wide variety of communications subsystems to be used. APPEL can also be specialised for completely new domains. For example, in a follow-on project funded by EPSRC it is planned to extend the core language to formulate management policies for wind farms.

- The specialisation of APPEL for call control deals with obvious events such as call setup and call failure. However, many other kinds of events are supported. For example, a common failing of other approaches like CPL is that they do not support mid-call events (e.g. adding a third party or a new medium) or end-call events (i.e. on termination). APPEL also reflects the increasing importance of presence and availability, which can trigger policies based on these. This allows integration with new services such as IM and ICQ, and with presence monitoring services such as supported by active badge systems. A policy trigger establishes values that may be used in conditions (simple facts such as the caller or callee, but also further information such as the location, role or capabilities of the call parties). Call actions include obvious ones like accepting, rejecting or forwarding a call. Other actions support modern communications services, e.g. forking (simultaneous call attempts) and recording presence/availability.
- To ease parsing and interchange of policies, APPEL uses XML. Two schemas have been defined and published on the web (*www.cs.stir.ac.uk/schemas*), one for call control and one for conflict handling.

2.3 Policy Server and Conflict Handling

The policy server stores, retrieves and enforces policies. Since APPEL is XML-based, it made sense to use an XML database. Although IBM TSpaces was the preferred choice for technical reasons, the policy store is accessed via a well-defined interface. This makes it easy to replace TSpaces by another form of database (e.g. an XML database or a more conventional relational one). TSpaces is named after its focus on tuple spaces – something that the policy system is able to exploit.

The policies retrieved for a call may come from a variety of sources: the caller, the callee, their domains, and any intermediate systems. The policy system is designed to be distributed. Multiple policy servers and multiple communications servers may be involved in a call. A conflict arises if the pool of policies dictates incompatible actions. Some action pairs may be benign (e.g. 'forward the call' vs. 'add video to the call'). Some action pairs are clearly incompatible (e.g. 'fork the call' vs. 'reject the call'). The design of the policy server [8] allows for a variety of policy conflict strategies: pre-negotiation (prior to making a call), post-negotiation (after making a call), offline (definition-time) conflict analysis, and online (call-time) conflict analysis. The project approach to conflict handling is described in [1, 7].

It was decided to focus on online rather than offline policy conflicts. Handling dynamic conflicts is technically much more demanding, partly because the environment is much less defined (any subscribers with any policies) and partly because there are real-time constraints. Static analysis is essentially a special and simpler case of dynamic analysis: the policy context is defined by the user's organisational setting. Rather than incorporate fixed conflict rules into the policy server (which would make it application-specific), conflict handling was externalised. Special resolution policies *define* the meaning of conflict, and *define* how to handle this. Straightforward conflict situations can be specified easily. However, more subtle conflicts can also be treated. For example, adding an external party to an international call might be considered incompatible for technical or economic reasons. Similarly, adding video to a call might be treated as incompatible for certain types of calls or for certain call parties.

As discussed in [1], dynamic handling of call conflicts raises many complex issues in a distributed setting. For example, forwarding or forking a call requires careful treatment. If the call routing is simply altered as a result, the resulting caller and callee policies may prove to be irreconcilable. This requires the policy servers to treat forwarding and forking actions as provisional. The policies associated with alternative routes are accumulated on a 'blackboard' (the policy store associated with the first policy server). The full set of policies can then be considered. This might lead to forwarding/forking being denied in order to resolve conflicts.

Conflict handling in call control has been able to exploit the availability of voice. When conflict causes a call to be rejected or certain policies to be suppressed, the user is informed by pre-recorded voice announcements (e.g. 'your call cannot be completed because the callee is currently unavailable'). Providing such information could be a breach of privacy or security, so it is defined by the resolution policies.

2.4 Policy Wizard

Policies can be written and uploaded in raw XML, but obviously this is only for specialists. A web-based policy wizard was therefore developed to help ordinary subscribers define policies. Two experimental policy wizards were created, one PHP-based and one servlet-based. The final version (documented in [10, 11]) is JSP-based. The advantages of a web-based wizard are familiarity, platform independence, and location independence.

Considerable efforts were made to ensure that the policy wizard was accessible to non-technical users. The wizard allows a near natural language version of policies to be edited. The wizard is multi-lingual, currently supporting English, French and German. Preliminary investigation of other languages (including Greek, Hungarian and Mandarin) showed that the approach is viable for most natural languages. The wizard user is given extensive online help, advice and hints. The most common kinds of policies are defined as templates. This allows frequently required policies to be created easily, simply filling in a few key parameters (e.g. the times between which a policy should apply, or the addresses to which calls should be forked). A policy system administrator can also define policies for groups of users or for inexperienced users.

2.5 Development and Demonstration

Work on theoretical and linguistic underpinning of policies was complemented by extensive development and demonstration. In order to show that policy-based call control was practical, it was essential to create an experimental setup. The project approach is protocol-independent. Support was demonstrated for policy-based control of conventional telephony, Internet telephony (SIP, H.323), PBXs (7000 ICS), softswitches (SER), and wireless devices (Blackberry, cellphone). This established that the approach is generic and of value in a wide variety of communications environments. The project approach is also platform-independent. Java was used throughout the development, enabling the software to be demonstrated on three different platforms (Solaris, Linux, Windows). The policy wizard was validated on a variety of web browsers and platforms.

A close link was maintained throughout with Mitel. Monthly audio/video-conferences were used to exchange technical ideas. This industrial collaboration was very valuable, as it ensured that the work was relevant and practical. Four visits were made to Mitel Canada for technical discussions and for demonstration of the evolving results. Mitel invited a number of other companies to attend these demonstrations. This led to project personnel being introduced to MKC Networks, adding a further and effective industrial relationship to the project.

It was originally hoped to create a Stirling-Mitel network for project demonstrations. However, practical considerations (firewalls and security restrictions at all sites) made this problematic. The solution was for Stirling to develop its software using exactly the same technical infrastructure as that available to Mitel: the 7000 ICS (Integrated Communications Server) developed by MKC Networks. This made code-sharing and integrated development possible.

3 Project Plan Review

The project proceeded substantially according to plan. All the key objectives were met, and in some respects exceeded. Early in the project, the global recession forced Mitel to close its Bellshill office. However, all technical and financial obligations were taken over by Mitel's corporate research centre in Canada, with whom the project developed a close working relationship. Attached to this report is a letter from Mitel confirming that their contributions to the project were substantially as expected.

The only technical obstacle was finding a suitable SIP platform with which to link the policy system. To some extent the project suffered from being an early adopter of SIP for research. The SIP standard itself was substantially revised during the second year of the project. Stable support of SIP proved difficult to obtain. Initially the project used research software developed by Columbia University. This was sufficient to make a start, but its limitations at the time made it unsuitable for the project. A number of other academic solutions were tried, but proved to be inadequate. Contact was made with various companies providing commercial SIP products. Unfortunately, these were either opaque (no suitable API) or were only building blocks for a SIP solution.

During the second year of the project, SER (SIP Express Router) was released by the Fraunhofer Institute. As an open-source project, this proved suitable for use by the project. A SER module was developed for use with the policy system. Through a separate KTP (Knowledge Transfer Partnership) with Edinburgh Network Technologies, Stirling also gained experience of the Vocal SIP softswitch.

Access to a commercial SIP solution was also obtained. During the second year of the project, MKC Networks brought to market their 7000 ICS (Integrated Communications Server) – a PBX that supports SIP phones. Mitel underwrote a very substantial discount on this PBX plus six SIP phones. The need for this could not have been anticipated at the time of the proposal. The Faculty of Management at Stirling met the discounted price of £2,250 for this equipment. Apart from providing a stable SIP base, the 7000 also led to direct collaboration with MKC Networks and to demonstration of project results on this platform as well as SER.

The project plan required a focus on SIP-based communication, but with H.323 also to be considered. Thanks to a postdoctoral grant from the Sino-British Fellowship Trust, Dr. Tingxue Huang visited Stirling for a year to work on policy-based control of H.323 systems using an open-source gatekeeper (GNU GK). At no extra expense to the project, this resulted in the integration of H.323 support with the policy work [3].

4 Research Impact and Benefits to Society

The project research has achieved results of international significance. Most work by others on SIP has been industrial and pragmatic (Columbia University being the main exception). Most work by others on policy-based management has not addressed the needs of call control (the University of Milan being the only other example known to the investigators). Policy-based call control for SIP and H.323 is a unique achievement of the project. The enthusiastic reception by Mitel and other companies has confirmed that the research has real commercial potential. As discussed in section 2.5, the approach is protocol-independent, platform-independent, and user-oriented. The project technical designs have been published for the benefit of the wider community [8, 9, 10]. Because of commercial potential, it is not planned to release the substantial amount of source code developed by the project. However, a binary release of the code is planned shortly for the benefit of the research community.

5 Explanation of Expenditure

Expenditure on the project was close to each budget figure, being slightly under-budget on travel/subsistence and consumables, and slightly over-budget on equipment.

Staffing It was planned to appoint an RA for three years. Dr. Stephan Reiff-Marganiec was the selected individual. However towards the second year of the project he decided to take up a lectureship at the University of Leicester, so he completed only two years on the project.

It might have been difficult to find a replacement RA for only one year. However the project was fortunate in finding two well-qualified individuals: Dr. Lynne Blair and Mr. Jianxiong Pang. With the agreement of EPSRC, the third year of the project was stretched to 18 months and was staffed by these two RAs working part-time. Dr. Blair holds a senior lectureship at Lancaster University, but obtained the agreement of her Department to spend a sabbatical year working on the project for Stirling. Because she retained teaching commitments at Lancaster, she was able to work only part-time on the project. Mr. Pang had just completed his PhD research at Lancaster, but was in the process of writing up. He was therefore able to work only part-time on the project. The contribution of Dr. Blair and Mr. Pang was equivalent to one RA full-time during the final year of the project. The tasks in the final phase allowed a ready division of effort between Dr. Blair and Mr. Pang.

As budgeted, Mr. G. Cochrane provided technical support throughout the project: web site maintenance, deployment and maintenance of specialised hardware and software. Also as budgeted, Mrs. G. McArthur provided clerical support for the project and its collaboration with Mitel and the three Canadian Universities.

Mitel Networks Corporation, as industrial sponsor, met their commitment to provide £21,000 in total to top up the RA's salary to industrial levels. This was crucial initially in attracting Dr. Stephan Reiff-Marganiec as an experienced researcher. Mitel also contributed staff time throughout the project: an assistance vice-president managed Mitel's involvement in the project, a consultant provided technical input throughout the project, and an engineer was actively involved in binding the project results to particular telephony platforms. Mitel estimate their staff time as equivalent to £35,000 over the whole project.

Travel and Subsistence As anticipated in the proposal, regular research visits were made to Mitel. Although the investigators kept in touch monthly with Mitel, face-to-face meetings were vital for effective collaboration and for demonstrating the evolving research. The project also presented its results at a number of key conferences: FORTE (Formal Techniques for Distributed Systems), ICFI (International Conference on Feature Interaction) and SDL (System Design Languages). On two occasions, it proved possible to combine visits to Mitel with conference attendance.

Equipment and Consumables A SUN workstation was purchased to act as the main policy server. Due to falling equipment prices, it was also possible to purchase a laptop to supplement this. An important aim of the project was to deliver platform-independent solutions. The SUN provided a central server running Solaris, while the laptop provided a development platform running both Windows and Linux. The laptop was regularly used for off-site demonstrations. Most of the expenditure on consumables was maintenance on the SUN and the laptop.

Mitel donated software and hardware to the project in the form of SIP software and PBX software. As noted in section 3, Mitel and the Faculty of Management supported the purchase of a SIP-based PBX and SIP phones. For the SIP/PBX software, Mitel provided free software licences, free technical support, and free upgrades throughout the project. In total, Mitel estimate their equipment support as equivalent to £10,000.

6 Further Research or Dissemination Activities

Research results were disseminated throughout the project, making use of journals and rigorously refereed conferences. A project website (*www.cs.stir.ac.uk/accent*) was maintained, with links to published papers and ongoing research activities. The project enabled Stirling staff to develop their expertise in policy-based system management. As a result, Stirling was able to bid successfully on a new project (joint with the Universities of Kent and Strathclyde). This EPSRC grant (Network of Distributed Sensors for Proactive Condition Monitoring of Wind Turbines, C014804) will support research by Stirling on policy-based management of wind farms.

Industrial collaboration was important throughout the project. Mitel introduced project personnel to two other companies: MKC Networks and New Heights Software Corporation. During visits to Mitel, the results of the project were disseminated and demonstrated to a wide industrial audience. A formal collaboration agreement with Mitel was signed early in the project. In return for their sponsorship and support, Mitel was granted a royalty-free, non-exclusive licence to turn the project results into commercial products. At the time of writing, two groups within Mitel are actively looking at this. MKC Networks and New Heights Software Corporation are also considering licensed use of the results in products. It is hoped that this will lead to new commercial developments starting within a year.

Separate from the Mitel collaboration, but with Mitel's knowledge, project personnel were involved in discussions with Iotum Corporation about licensing and consultancy on policy-based call control. It is hoped that these ongoing negotiations will lead to new commercial opportunities within a year.

Project personnel were invited to join an international working group working under the auspices of IEEE on policy-based management [2]. The project also stimulated separate research on modelling and analysing new kinds of services: SIP services, voice services, and home network services. A number of publications, not cited here, resulted from this research. Although this work was orthogonal to the project, the two were in synergy. It is likely that this additional research will bear fruit in the form of other grants and industrial collaborations.

References († means accepted for publication and in press)

[1] L. Blair and K. J. Turner. Handling policy conflicts in call control. In S. Reiff-Marganiec and M. D. Ryan, eds. *Proc. 8th. Feature Interactions in Telecommunications Systems*, IOS Press, Mar. 2005. †

[2] P. Dini, A. Clemm, T. Gray, F. J. Lin, L. Logrippo, and S. Reiff-Marganiec. Policy-enabled mechanisms for feature interactions: Reality, expectations, challenges. *Computer Networks*, 45(5):585–603, Aug. 2004.

[3] T. Huang and K. J. Turner. Policy support for H.323 call handling. *Computer Standards and Interfaces*, Jan. 2005. †

[4] S. Reiff-Marganiec. Policies: Giving user control over calls. In M. D. Ryan, J.-J. C. Meyer, and H.-D. Ehrich, eds. *Objects, Agents and Features*, LNCS 2975, pp. 189–208. Springer, May 2004.

[5] S. Reiff-Marganiec and K. J. Turner. Use of logic to describe enhanced communications services. In D. A. Peled and M. Y. Vardi, eds. *Proc. Formal Techniques for Distributed Systems*, LNCS 2529, pp. 130–145. Springer, Nov. 2002.

[6] S. Reiff-Marganiec and K. J. Turner. A policy architecture for enhancing and controlling features. In D. Amyot and L. Logrippo, eds. *Proc. 7th. Feature Interactions in Telecommunications Systems*, pp. 239–246. IOS Press, June 2003.

[7] S. Reiff-Marganiec and K. J. Turner. Feature interaction in policies. *Computer Networks*, 45(5):569–584, Aug. 2004.

[8] S. Reiff-Marganiec and K. J. Turner. The ACCENT policy server. Technical Report CSM-164, Computing Science and Mathematics, University of Stirling, UK, May 2005.

[9] S. Reiff-Marganiec, K. J. Turner, and L. Blair. APPEL: The ACCENT project policy environment/language. Technical Report CSM-161, Computing Science and Mathematics, University of Stirling, UK, May 2005.

[10] K. J. Turner. The ACCENT Policy Wizard. Technical Report CSM-166, Computing Science and Mathematics, University of Stirling, UK, May 2005.

[11] K. J. Turner, S. Reiff-Marganiec, L. Blair, J. Pang, T. Gray, P. Perry, and J. Ireland. Policy support for call control. *Computer Standards and Interfaces*, Jan. 2005. †