A Personalisable Internet Telephony Service

David Lesaint and **John Ly**[1]
and **Deepak Mehta** and **Barry O'Sullivan** and **Luis Quesada** and **Nic Wilson**[2]

1 Introduction

Information and communication services are playing an increasing and potentially disruptive role in our lives. As a result, service providers seek personalisation solutions to allow users control and enhance the way services are delivered. Call control in particular has received much attention in the telecommunications domain. An outcome of this work has been the emergence of features as fundamental primitives for personalisation.

A feature is an increment of functionality that modifies the basic service behaviour, e.g., call-screen, call-divert, find-me, multimedia ring-back tone, voice-mail, mid-call-move. Features are optional and must be activated to fulfill their role. Once activated, they execute automatically or interactively during sessions. Service personalisation then refers to the problem of selecting which features should be active and when.

Our project[3] investigates the application of Constraint Technology to support this task. Our demonstrator shows how users can configure their feature subscription through a web-based interface with the assistance of a constraint engine. It also shows how features are automatically activated during voice-over-IP sessions consistently with the subscriptions pre-configured by participants.

2 Feature Subscription Configuration

Features are fine-grained and address specific concerns such as privacy, mobility, etc. Consequently, service providers organise their feature catalogues to ensure minimum redundancy and maximum combinability between features. Feature composition is a source of interactions though. A feature interaction is "some way in which a feature modifies or influences the behaviour of another feature in generating the system's overall behaviour" [1]. For instance, a do-not-disturb feature blocks incoming calls and may cancel the effect of other features, e.g., a welcome announcement. Users may consider such interactions desirable or undesirable depending on the scenario.

Our approach to address feature interactions is based on Distributed Feature Composition - an abstract service architecture designed for feature composition and interaction management [3][1].

In DFC, configuring a feature subscription involves selecting, parameterising and sequencing features consistently with predefined feature interaction constraints and integrity rules. In DFC each feature is implemented by one or more modules called *feature box types* (FBT). Here we assume that each feature is implemented by a single FBT and we associate features with FBTs. A call session between two end-points is set up by chaining feature boxes, i.e., instances of FBTs. The routing method decomposes the connection path into a source and a target region and each region into *zones*. A source (target) zone is a sequence of features which execute for the same source (target) address.

Fig. 1 shows a toy catalogue comprising features for callers (e.g., originating-call-screening - OCS) and callees (e.g., terminating-call-screening - TCS -, time-dependent-routing - TDR). The catalogue also prescribes binary precedence and exclusion constraints between features, e.g, call-log (CL) must precede TDR (CL<TDR) in any subscription. The first source zone is associated with the source address encapsulated in the initial setup request, e.g., zone of X in Figure 1. A change of source address in the source region, caused for instance by an identification feature, triggers the creation of a new source zone. If no such change occurs in a source zone and the zone cannot be expanded further, routers switch to the target region. Likewise, a change of target address in the target region, as performed by Time-Dependent-Routing (TDR) in Figure 1, triggers the creation of a new target zone. If no such change occurs in a target zone and the zone cannot be expanded further (as for Z in Figure 1), the request is sent to the final box identified by the encapsulated target address.

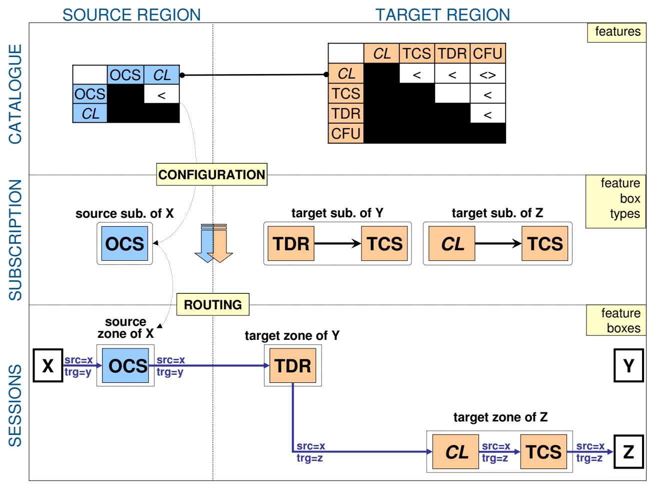

Figure 1. From feature catalogues to subscriptions and sessions

[1] BT Research & Venturing, UK
[2] Cork Constraint Computation Centre, University College Cork, Ireland
[3] The BT/4C collaboration is supported by Science Foundation Ireland and the Embark Initiative.

3 Functionality

Fig. 2 shows a snapshot of the widget-based web interface to the subscription configuration system. The interface exposes the catalogue from which users can pick and sequence features through drag-and-drop operations. The constraint engine then guides the user towards consistent subscriptions in different ways:

verification - checking the consistency of the input subscription;
partial completion - if consistent, extend it with entailed precedence constraints;
filtering - if consistent, filter out the set of additional features and precedence constraints that would make it inconsistent;
completion - if consistent, suggest complete and consistent extensions;
revision - if inconsistent, suggest consistent and optimal relaxations.

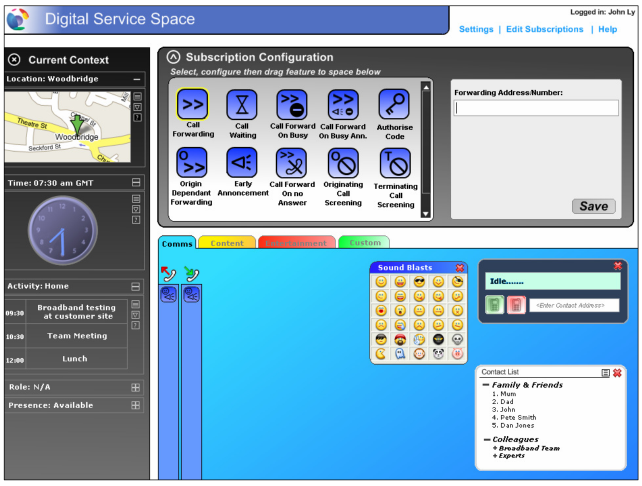

Figure 2. Subscription interface

These services are implemented by modeling and solving the feature subscription configuration problem as a combinatorial problem [5]. The constraint engine implements branch-and-bound and constraint filtering algorithms to compute relaxations and graph algorithms (topogical sort and transitive closure) for the other tasks. It is programmed with the Java-based constraint programming library Choco [4]. Alternative approach is to compile the catalogue in to a binary decision diagram in the offline phase and based on the input of a user use polynomial time operations in the online phase [2]. The subscription interface is implemented using JavaScript, Java Server Pages and Ajax techniques. The service delivery platform is itself based on the Session Initiation Protocol [6][7] with routers and features implemented as SIP servlets.

REFERENCES

[1] Gregory W. Bond, Eric Cheung, Hal Purdy, Pamela Zave, and Christopher Ramming, 'An Open Architecture for Next-Generation Telecommunication Services', *ACM TOIT*, **4**(1), 83–123, (2004).
[2] Tarik Hadzic, David Lesaint, Deepak Mehta, Barry O'Sullivan, Luis Quesada, and Nic Wilson, 'A BDD Approach to the Feature Subscription Problem', in *Prestigous Applications of Intelligent Systems (PAIS)*, Patras, Greece, (July 2008). To appear.
[3] Michael Jackson and Pamela Zave, 'Distributed Feature Composition: a Virtual Architecture for Telecommunications Services', *IEEE TSE*, **24**(10), 831–847, (October 1998).
[4] F. Laburthe and N. Jussien. JChoco: A java library for constraint programming.
[5] David Lesaint, Deepak Mehta, Barry O'Sullivan, Luis Quesada, and Nic Wilson, 'Personalisation of Telecommunications Services as Combinatorial Optimisation', in *IAAI-08*, Chicago, USA, (2008). AAAI Press. To appear.
[6] Jonathan Rosenberg, Henning Schulzrinne, Gonzalo Camarillo, Alan B. Johnston, Jon Peterson, Robert Sparks, Mark Handley, and Eve M. Schooler, 'SIP: Session Initiation Protocol', RFC 3261 (standard), IETF, (June 2002). Updated by RFCs 3265, 3853, 4320.
[7] Robert Sparks, 'SIP: Basics and Beyond', *ACM Queue*, **5**(2), 22–33, (March 2007).