EUROPEAN TRANSACTIONS ON TELECOMMUNICATIONS
Euro. Trans. Telecomms. 2004; **15**:1–13 (DOI: 10.1002/ett.990)

Preference-based session management for IP-based mobile multimedia signaling[†]

Wolfgang Kellerer[1]*, Matthias Wagner[1], Wolf-Tilo Balke[2] and Henning Schulzrinne[3]

[1] *Future Networking Lab, DoCoMo Communications Laboratories Europe GmbH, Germany*
[2] *Electrical Engineering and Computer Science, University of California at Berkeley, USA*
[3] *Department of Computer Science, Columbia University, New York, NY, USA*

SUMMARY

The increasing variety of mobile multimedia services raises the need for mechanisms to select those services whose capabilities best match individual user requirements. In this paper, we present an advanced concept for service personalization in next-generation IP-based mobile systems based on the notion of user preferences. The described concept features several enhancements for personalization, including the intuitive modeling of user preferences, the construction of complex preference terms from basic preference expressions and a stepwise refinement of profiles and preferences. We describe this solution applied to the signaling mechanism of the session initiation protocol (SIP), which is the protocol suite for application control in current IP-based mobile communication systems such as UMTS. In particular, our preference concept for personalized service management extends the conventional SIP preference modeling methods proposed so far: a preference order on the available service features is introduced that accounts for the specific demands of a user without a numerical ranking of features. We show how base preferences can be intuitively combined to complex preference expressions and explain how these relate to standard SIP feature preferences. Copyright © 2004 AEI.

1. INTRODUCTION

Personalization is regarded to be one of the most compelling features of future mobile communication systems. User-centered services and personalization promise to support customers in selecting their favorite services from the rapidly increasing diversity of mobile multimedia services and adjusting their services to their individual needs. By services, we refer to end-user services and applications such as telephony, conferencing, information retrieval and entertainment (e.g. gaming). We consider personalization as matching of a user's preferences and demands to the available services under the constraints of a given situation or environment. To select and tailor services to the actual demands, we need to take into account [7]:

- knowledge about users and their context,
- the capabilities of available services,
- and the capabilities and constraints of the network employed.

Given the diversity and highly dynamic nature of mobile communication systems, for example concerning heterogeneous networks and terminals, personalization is not only important for the discovery and selection of services [2, 3], but also for establishing and managing service sessions on behalf of an individual user. In traditional multimedia communication systems the network signaling system only supports the end-to-end transport of basic service capabilities and user preferences, typically expressed as simple parameter or feature sets. The negotiation process itself is performed by the applications. Moreover, service selection is performed off-line and the resulting service is bound to a network address of a specific server, for example by picking up a servers IP address from a search. Systems for service discovery such as Jini [20] or

---

* Correspondence to: Wolfgang Kellerer, Future Networking Lab, DoCoMo Communications Laboratories Europe GmbH, Germany.
E-mail: kellerer@docomolab-euro.com

[†] This paper is partly based on work presented at the 8th International Workshop on Mobile Multimedia Communications (MOMUC2003), Munich, Germany, October 2003.

Received ??????
Revised ??????

the service location protocol [8] give support to an automatic, profile-based service selection. However, these solutions only scale to local networks and do not consider service routing/selection as described in this paper. Moreover, the trend towards global interworking and the increasing need for context aware applications [15] that go beyond location-based applications, cause that the parameter sets to be processed will soon become very complex. To cope with this complexity, advanced user support and application support is needed from within the communication network.

In this paper we propose an extension of our work in Reference [11] as an advanced approach to a preference-based management of service sessions for IP-based mobile multimedia systems. According to existing approaches for user preference and capability descriptions, such as Reference [17], we model service parameters as feature predicates in first-order predicate logic. In addition, we allow users to express their personal wishes and dislikes more naturally in terms of a preference order on these feature predicates. Presenting a sample call center scenario, we show how to leverage the personalized session management by user preferences that are already pre-processed in the network. In general, preferences could be used to support:

- service selection in a service catalogue or service portal;
- service request pre-processing in network entities;
- service selection performed in the network based on matching of preferences and available service capabilities;
- advanced capability negotiation to adapt and customize the selected service, for example on the server;
- efficient propagation of profile information through the network.

Note, that a service could be any communication endpoint including a user terminal or for instance a server hosted video. The same service might be registered with different names reflecting different capability variants, for example when a user is reachable at different terminals or a video is available in various resolutions.

IP-based mobile service architectures, to which our approach applies, have already been specified in detail for the third generation mobile communication systems. In this respect, the most important example for IP-based mobile networks is the IP-based Multimedia Subsystem (IMS) standardized by the 3GPP for IMT2000 [1]. 3GPP has adopted the session initiation protocol (SIP) of the Internet Engineering Task Force [19] to serve as the application layer signaling protocol in the IMS.

In the following, we will focus on the network support of preference-based service management and in particular on service selection. Our system enhancements include solutions for service selection based on a sound preference model and preference handling algebra. It also covers aspects like efficient profile propagation. Furthermore, we describe the application of our approach in an advanced signaling architecture for IP-based mobile communication networks based on the Session Initiation Protocol (SIP) [19]. Our concept is not limited to mobile systems, but can also be applied for all kinds of multimedia communication systems. As an extension to SIP, Rosenberg *et al.* [17] proposes a method for preference handling in SIP that has already been discussed for quite some time in the community. However, we still see improvements regarding the preference model, which we will lay out in detail in the course of this paper. So, in this work we are using the specification given in Reference [17] as a basis as it describes the main state of the art in preference handling for IP multimedia.

The remainder of this paper is structured as follows: first we describe the possibilities and advantages of network-supported preference handling with our target conceptual framework. In Section 3, we give details about the current preference modeling and handling in SIP. The advantages and mechanism of our proposed preference model compared to the latter are described in Section 4. A detailed example illustrates our solution, which is running through all following sections. In Section 5, we describe more advanced issues in preference handling such as feature set matching, negative preferences and implicit preferences. Efficient propagation of profiles is worked out in Section 6. Before we conclude this paper with a summary, Section 7 gives an overview over state of the art concerning preference-based frameworks.

2. CONCEPTUAL FRAMEWORK

In this section, we focus on architectural issues and show the targeted application of preference-based service management in an IP-based mobile multimedia signaling architecture. Moreover, we illustrate the roles of the network entities in an efficient preference-based service management.

2.1. IP-based mobile multimedia service architecture

In a typical mobile communication network signaling messages, which possibly include the above described preferences, traverse several functional entities in the communication path between client application and server application such as a proxy in a visited network or the home network session manager. Each of these entities maintain different information that can be used to process a preference-based service request more efficiently (cf. Figure 2). Examples include:

- the *user device* stores part of user profile and preferences, especially the applicable preferences with respect to the device capabilities;
- The *access node* transports capabilities, current workloads or service guarantees (e.g. in terms of bandwidth) and the user context.
- *Visited network (VN) proxy server*: local information, here a decision can be taken to select local services or home network based services (without knowing the detailed user profile).
- The *home network (HN) gateway* manages the access to an operator's domain from a foreign domain. It may host network based user profiles and handles user authorization.
- *Home network session manager*: manages service profiles. Here a decision is to be taken to select an adequate server/called party serving the requested service

These are basic entities in mobile multimedia communication architectures such as the IMS described above. Here, the use of SIP provides several capabilities that we use for preference processing. SIP uses an overlay addressing scheme. SIP addresses are different from the IP-network addresses, i.e. the target endpoint address is decoupled from the network address. This allows selecting or adapting the request route while the request is traversing several signaling entities such as the above.

Furthermore, SIP allows modifying the request content in traversed network entities acting as SIP proxy servers. This includes preferences according to Reference [18] that are part of the request message syntax or the session description that is carried as additional payload independently of the SIP request. In this way, preferences can be updated or modified in the network entities such as proxy servers.

2.2. *Proxy-based signaling*

The underlying principle of a proxy-based next generation application layer signaling architecture is described in more detail in References [10 and 9] focusing on a SIP-based transaction protocol for signaling for session management. There, network servers such as access session controller, service session controller and communication session controller are realized as SIP proxy servers or SIP redirect servers that are traversed by session signaling messages. This architecture is easily transferred to the IP-based architecture that we take as a basis for the considerations described here.

In the architecture of Reference [9], two features are described that we use in the following: the selection of the respective servers and redirecting the signaling messages accordingly, and the stepwise refinement of the session description (e.g. adding service profile) when it is processed by the proxy servers. In this architecture, the end-to-end concept of IP-based signaling is kept and at the same time one makes use of the information or control intelligence that is available in the network. In Reference [9] user profiles have been considered as part of the session description, but their effect on session management has not been investigated yet.

2.3. *Motivating example*

We take a call center session to illustrate our concepts. The profiling issues proposed in this paper are considered together with the different functionalities of the involved network components (cf. Figure 2).

A system expert, who is currently working at a customer's site, wants to get further technical details about the installed system he is working on. Therefore, he is contacting the manufacturer's call center to receive support including multimedia information in form of live videos. The connections are running over a wireless link. He sets his personal preferences concerning the desired languages of the call center assistant and the preferred multimedia presentation features (e.g. video codec, video resolution) among other parameters not shown here. As these preferences are regarded as soft constraints he also provides his opinion about second choice capabilities as will be described in more detail in the following sections. See especially Section 4 for the preference descriptions. For the following steps of our example, we give references to further sections where the respective issues are discussed in more detail.

The preferences are transmitted in the service request (step1) over the access network to the core network of the currently visited network domain. On passing a proxy in the access network, some preferences can be singled out (step 2), for example given that limited wireless capabilities do not allow for transmitting high resolution videos, and respective constraints are added (cf. Section 6.1). Since local information servers of the visited network domain (step 3) cannot provide the requested content, the request is routed to the user's home network domain (cf. Section 6.3). There the request is matched with default preferences (cf. Section 6.2) stored on the home network profile server (step 4), which also might add some typical (implicit) preferences that have not explicitly been specified by the user in the individual request (cf. Section 5.5).

The home network session manager (step 5) serves as the point for service selection by mapping the user's remaining preferences and the constraints added by network nodes with the capabilities of available services (such as end users devices) to select a suitable server providing the requested information (cf. Section 5.1).

Fortunately for our sample user, a MPEG-capable server for the requested information could be found. The remaining preferences and constraints are now transmitted to the server to customize the service. Here, also further optional capabilities can be selected without necessary involvement of the user in the negotiation process.

In the following sections, we describe our solution regarding the above mentioned mechanisms in detail, compared to the current IETF approach for SIP, where applicable, and backed by a detailed running example.

3. PREFERENCE HANDLING IN SIP

In addition to RFC 3261, which specified the SIP [19], the IETF SIP working group has drafted an extension to SIP to support the so-called caller and callee preferences. Such preferences can be used for profile-based service request negotiation and routing as well as capability matching when a request reaches an application server [17, 18]. As motivated above, in the following we will focus our considerations on preferences in their most intuitive form of soft constraints. However, if a user feels that for some reason a preference is definitely needed, it can also be formulated as a hard constraint.

Caller preferences are needed in multimedia communications since a large number of devices can register as endpoints for a single user address. The respective proxy server selects a subset of these devices and contacts them sequentially or in parallel. This operation is called sequential or parallel forking. Caller preferences allow the caller to express a preference among the multitude of callee devices and could help to avoid calls to less preferable devices. Some of the negotiations, in particular for media capabilities, could of course already take place as part of the conventional call setup, for example by inspecting the session description contained in the invite request. Non-compatible devices would then reject the call and only compatible devices would ring.

Caller preferences allow pushing preference logic into the callee's proxy server, allowing for appropriate sequencing of call attempts to devices in decreasing order of preference, which a pure end-system approach would permit. For wireless systems, we thus avoid the overhead of signaling over wireless last hop links.

Reference [18] uses 'Contact' and 'Accept-Contact' headers to describe callee capabilities, describing the features of one registered device at a callee's proxy server (such as the home network session manager in our scenario), and caller preferences that are matched against those capabilities. In this way, 'Accept-Contact' headers are used to select the best matching communication endpoint, which is described by its capabilities stored as 'Contacts' in network proxy servers, such as for example in step 5 of our example.

Intuitively, the preferences used for an enhanced negotiation have to be understood as wishes that, however, cannot always be fulfilled. In that sense, preferences indicate feature constraints that a session should fulfill to best meet its requirements. On the other hand, even if none of the preferences are met, a session initiation should be possible. In terms of the described SIP extension this restricts us to SIP 'Accept-Contact' headers without any 'require' tags as defined in Reference [17]. We will briefly revisit and discuss further preference modeling options for SIP in Section 5.

Table 1 provides a basic idea of how preferences are coded and incorporated into a SIP request to indicate preferences among service capabilities. A caller can add one or more 'Accept-Contact' header fields to his request. Each Accept-Contact contains a set of feature parameters that define a feature set [13]. Multiple feature tags in a contact are connected by the operator ';' which is to be interpreted as logical conjunction. For instance Accept-Contact AC1 (the name is not part of the actual request header) indicates a preference for a session that allows for videos encoded in 'mpeg' or 'h261' at a 'high' resolution using the language French (fr) for conversation. Numeric 'quality values' (or 'q-values') [13] can be assigned to individual Accept-Contacts to express a preferential order.

Let us consider a SIP request that includes the Accept-Contacts AC1-AC4 and an assignment of q-values 1.0, 0.8 and 0.7 as given in Table 1. The left hand side of Figure 3 depicts the preferential ordering of Accept-Contacts for our example, for example a 'high' resolution video session of type 'mpeg' or 'h261' is preferred over a 'low' resolution 'mpeg' session (q-val(AC1) > q-val(AC3)). On the other hand, there is equal preference for the choice between 'high' resolution video sessions in either 'mpeg',

Table 1. Sample SIP Accept-Contact headers and SIP session contacts.

Name	Accept-Contact	q-val
AC1	`*;type="video/mpeg,video/h261";description="<high resolution>";languag820e="fr"`	1.0
AC2	`*;type="video/quicktime";description="<high resolution>";language="fr"`	1.0
AC3	`*;type="video/mpeg;description="<low resolution>";language="de"`	0.8
AC4	`*;type="video/mpeg;description="<low resolution>";language="jp"`	0.7
AC1′	`*;type="video/mpeg";description="<high resolution>";language="fr"`	1.0
AC1″	`*;type="video/h261";description="<high resolution>";language="fr"`	1.0
AC5	`*;type="video/h261";description="<low resolution>";language="fr"`	0.8

Name	Contact
C1	`sip:u1@h.example.com;type="video/mpeg;description="<low resolution>";language="de"`
C2	`sip:u2@h.example.com;type="video/quicktime";description="<high resolution>";language="fr"`
C3	`sip:u2@h.example.com;type="video/h261";description="<low resolution>";language="jp"`

'h261' or 'quicktime' (q-val(AC1) = q-val(AC2)). According to the q-value of AC4, the least preferred session in this example would be an 'mpeg' session at low resolution.

Multiple Accept-Contacts in a single SIP request are to be understood as logical disjunctions of possible sessions features, i.e. a session that either features AC1, AC2, AC3 or AC4 will eventually be initiated (if at all). Note that AC1 is the only Accept-Contact so far that contains a conjunction of preferred features, namely video in 'mpeg' or 'h261'. Thus, a refinement of AC1 that results in a split of AC1 into AC1′ and AC1″ as given in Table 1 is possible. Without a modification in the q-value assignments, i.e. q-value(AC1) = q-value(AC1′) = q-value(AC1″), the semantics of the request remain unchanged. The right hand side of Figure 3[Q7] illustrates this refinement. For the sake of the example the refinement of AC1 is complemented by a newly introduced Accept-Contact header AC5 indicating that 'low' resolution videos in 'h261' are a equal choice to the same resolution in 'mpeg' (q-value(AC3) = q-value(AC5)).

During session negotiation SIP Accept-Contacts headers are matched against available SIP session contacts to determine the best caller-callee match for the session. As we will see in the following, our example so far (with sample contacts given in the lower part of Table 1) will result in a match of AC2 to C2 as the best possible match.

4. TOWARDS ENHANCED PREFERENCE FRAMEWORKS

4.1. Problems of preference handling

Coded in q-values the modeling of preferences is currently rather limited in SIP: numeric values in the range of [0,1] with up to three fractional digits are allowed to rank feature sets. As we have seen, higher values are preferred over lower values and equal values are presumed to be equally preferred. Arithmetic operations on preference values are likely to produce unpredictable results and are thus generally not used. Furthermore, the assignment of q-values is restricted to full feature sets (complete Accept-Contacts) and not allowed at the feature level.

On the other hand, people rather state their wishes and dislikes in terms like 'I personally prefer A over B', 'I like all of kind A except for B' and so forth, instead of inventing numerical values to express a total ordering. This kind of preference modeling is universally applied and intuitively understood by everyone. Moreover, some natural combinations simply cannot be expressed by the limited capabilities of the numerical model. Thinking of preferences in terms of 'better than' on the other hand, also has a very natural counterpart in mathematics: such basic real life preferences can directly be mapped onto (strict) partial orders [6] in a straightforward manner. Q7

4.2. Preference model

We advocate that personalized session negotiation can benefit from an advanced preference model and therefore, propose to express user preferences as a (partial) ordering of feature predicate without the use of explicit quality or ranking values. Instead of numeric comparison or even manipulation of weighting factors, we propose to directly handle preferences as (partially) ordered feature sets as proposed for example in preference algebra. For instance, Reference [12] proposes a preference framework tailored to standard database systems together with a direct mapping to relational algebra and declarative query languages. This preference model is based on strict partial order

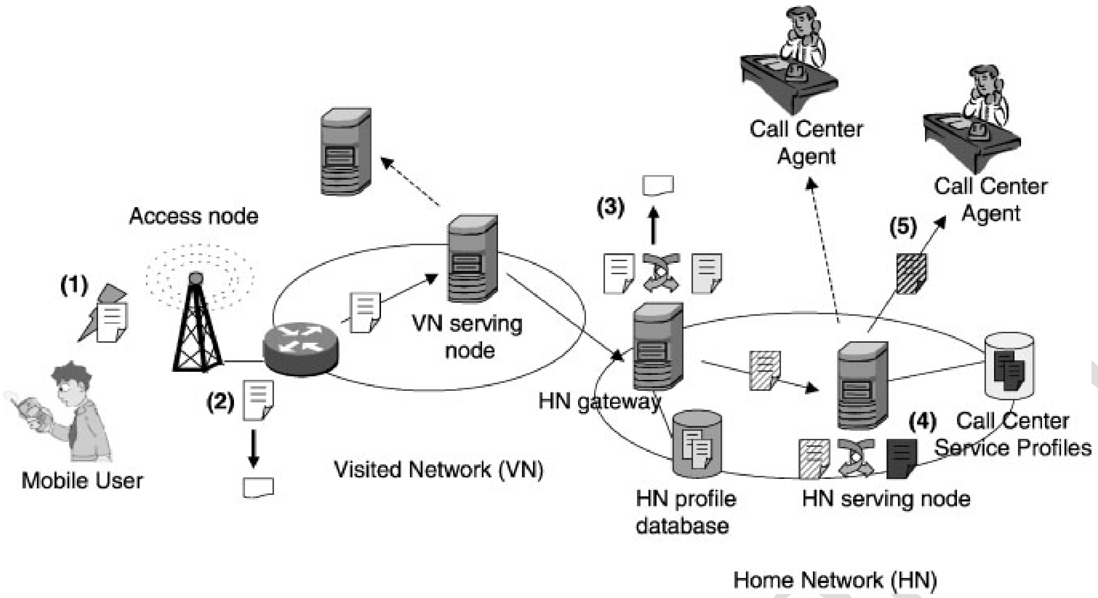

Figure 1. Proxy-based preference propagation and management.

semantics and features a variety of preference constructors that are intuitively combined to build complex preference expressions from base preferences. In this framework, a preference P is defined as a strict partial order $P = (A, <P)$, where A is a set of feature attributes and $<P \subseteq \mathrm{dom}(A) \times \mathrm{dom}(A)$ an order relation with dom(A) as the domain of feature A. $<P$ is irreflexive and transitive.

In the following, we will show how an approach similar to Reference [12] can be used to replace the limited SIP preferences. We start with base preferences for our running example as given in Figure 1. As we can see from the definition above, preferences are not applied to feature sets of multiple features, i.e. full Accept-Contact headers. Instead they are defined for each single SIP session feature: in our example a ‘video’ in ‘mpeg’ is preferable to ‘h261’ or ‘quicktime’ with an unbiased choice between ‘h261’ or ‘quicktime’. Analogously, ‘lang’ in ‘fr’ is considered to be superior to ‘de’, ‘en’ is considered an equally good choice with all of these language selections preferable to ‘jp’. This could be the language selection preference for someone who speaks French and German almost equally well (with the tendency to select French), speaks English anyway and prefers all of theses languages over Japanese. A member of DoCoMo Euro-Labs could have such a preference. In addition, naturally, ‘high’ video resolution is always preferred over a ‘low’ one.

Preference/capability matching can now be performed along the lines of the given preference order, for example matching the preferences from Figure 2 onto the capabilities in Table 1 can result in a call center session with videos encoded in ‘quicktime’ at ‘high’ resolution, with French language capabilities as a best possible match. As with q-values, these preferences are still treated as soft constraints during session negotiation, i.e. ‘mpeg’ is selected if available, otherwise ‘h261’ and ‘quicktime’ are treated without preference.

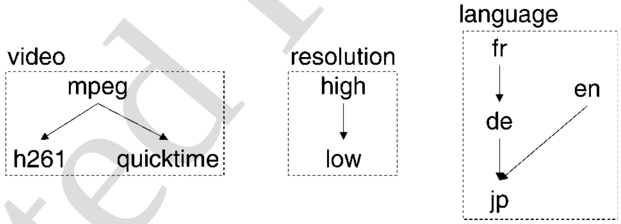

Figure 2. Individual caller preferences.

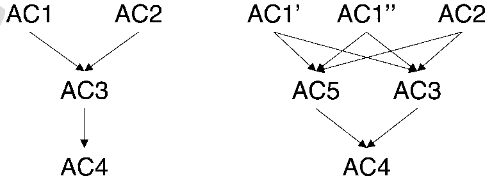

Figure 3. Order of Accept-Contact headers.

4.3. Preference construction

To manage SIP caller and callee preferences as partial orders of feature predicates a powerful and flexible modeling technique is needed. According to Reference [12], complex preferences can be inductively constructed from a set of suitable base preferences by means of preference constructors and complex preference combination. In the following, we will briefly discuss two basic combination operators, namely Pareto accumulation and preference prioritization, as possible means of preference resolution in multimedia session signaling systems such as SIP.

4.3.1. Pareto accumulation. The Pareto-optimality principle has been applied and studied intensively for decades for multi-attribute decision problems in the social and economic sciences. In our case it can be used to handle equally important feature preferences. We use the binary Pareto operator '$\otimes$' that is intuitively defined such that a matching feature tuple $v = (v1, v2)$ can only be preferred to another match $w = (w1, w2)$, if v is better than or equally good w in every single feature value with 'strictly better' in at least one value. Formally; let $P1 = (A1, < P1)$ and $P2 = (A2, < P2)$ be two preferences. For $x = (x1,\ x2)$ and $y = (y1,\ y2)$ with $x,\ y \in \mathrm{dom}(A1) \times \mathrm{dom}(A2)$ we define $x < \ (P1 \otimes P2)\ y$ iff $(x1 < P1\ \ y1 \wedge (x2 < P2\ \ y2 \vee x2 = y2)) \vee\ (x2 < P2\ \ y2 \wedge (x1 < P1\ \ y1 \vee x1 = y1))$. Therefore $P = (A1 \cup A2,\ P1 \otimes P2)$ is a strict variant of the coordinate-wise order of the Cartesian product [6] and is called a Pareto preference or the Pareto accumulation of P1 and P2.

Let us revisit the running example and consider the case where we want to further differentiate between the video codecs 'mpeg', 'h261' and 'quicktime' as shown in Figure 1, i.e. 'mpeg' is preferred over 'h261' and 'quicktime'. Using the SIP q-value approach, to express this video preference and to relate it with the preference on the video's resolution ('high' preferred over 'low', cf. Figure 1) according to Pareto semantics we would have to consider six combinations of 'video' and resolution and assign q-values accordingly. In Table 1 this could be done by introducing a new Accept-Contact, say AC6, and adjusting the quality values so that perfect matches in both, video and resolution, are considered best matches. The second choice would be matches in one session feature only followed by those that represent second grade matches. An example for a valid q-value assignment would thus be q-value($AC1'$) = 1.0, q-value($AC1''$) = q-value(AC2) = q-value(AC3) = 0.9, q-value(AC5) = q-value(AC6) = 0.8.

Note that with the standard q-value approach to SIP preference assignments, any combination of feature predicates has to be addressed separately to express and distinguish feature preferences. In contrast, with preference constructors such as '$\otimes$' we can express the same semantics (in this case Pareto-accumulation) in a single preference expression. Figure 4 shows the result of 'resolution$\otimes$video' that yields a preference ordering according to the above introduction of new Accept-Contacts, re-assignments of q-values and the treatment of matches in the features 'video' and 'resolution' without the use of a preference constructor. Note that this preference accumulation allows a very natural handling of incomparable service attributes: since 'h261' and 'quicktime' were considered incomparable in the video preference from Figure 2 any pairwise tuples of 'resolution$\otimes$video' that contain 'h261' and 'quicktime' are also considered incomparable.

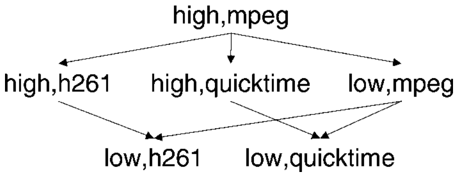

Figure 4. Pareto accumulation resolution$\otimes$video.

4.3.2. Preference prioritization. In the matching of service requests and capabilities, often some preferences might be considered more important than others. In addition to the equal treatment of preferences by Pareto accumulation, Reference [12] defines the preference operator '&' for preference prioritization. Intuitively, if 'P1&P2', i.e., preference P1 is prioritized over P2, then P1 is considered more important than P2. As a consequence there is no compromise in feature matching by P1. P2 is evaluated only where P1 gives several alternatives of equal usefulness, for example in the case of the choice between 'quicktime' and 'h261' codecs. Formally; let $P1 = (A1, < P1)$ and $P2 = (A2, < P2)$ be two preferences. For $x = (x1,\ x2)$ and $y = (y1,\ y2)$ with $x,\ y \in \mathrm{dom}(A1) \times\ \mathrm{dom}(A2)$ we define $x < (P1\ \ \&\ \ P2)\ \ y$ iff $(x1 < P1\ \ y1) \vee (x1 = y1 \wedge x2 < P2\ \ y2)$. In many practical cases this definition still holds even if $x1 \sim y1$, i.e. in the case where x1 and y1 are incomparable. Therefore, $P = (A1 \cup A2,\ P1\ \&\ P2)$ is a strict variant of the lexicographic order of the Cartesian product [6] and is called a prioritized preference.

In our running example, let us assume that a caller wants to express that the resolution of a video is actually more important to him than the video codec. Respecting the individual feature preferences for 'resolution' and 'video' this prioritization can be stated in the single preference expression 'resolution&video'. Figure 5 depicts how 'resolution&video' can be expanded for matching. As we demanded for videos in high resolution are generally

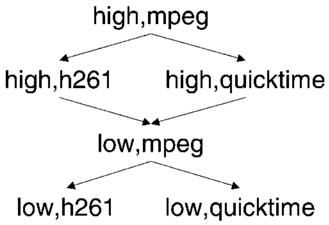

Figure 5. Preference priorization resolution&video.

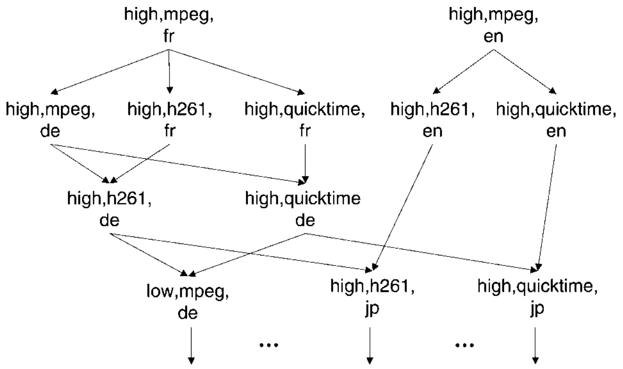

Figure 6. Complex preference (resolution⊗video)&language.

preferred over videos in low quality. Thereby, the base preference for video codecs, i.e. ‘mpeg’ preferred over ‘h261’ etc., is respected. Please note that opposed to the basic preference of ‘mpeg’ over ‘quicktime’, the prioritization of the resolution here leads to the preference of a ‘high, quicktime’ service over ‘low, mpeg’.

4.3.3. Complex preferences. Figure 6 gives an example of a complex preference, namely (resolution&video) ⊗ language. According to our discussion from above, this preference term states that a video’s resolution is of primary concern compared to its codec. However, the respective possible combinations of these features are equally important as the language encoding.

Note that, according to our earlier considerations, the incomparable attribute combinations are again respected, i.e. any pairwise tuples containing ‘h261’ and ‘quicktime’ remain incomparable (same for pairwise combinations of ‘fr’/‘en’ and ‘de’/‘en’ due to the language preference in Figure 2). Although Figure 6 is not showing all possible attribute tuples we can already grasp the complexity we would need to handle the preference (resolution&video)⊗language in the notation currently proposed for SIP: the number of ‘Accept-Contact’ headers would have to be at least equal to the number of nodes shown in Figure 6.

5. ADVANCED SIP PREFERENCE HANDLING

In this section, we briefly discuss further extensions of SIP as proposed in Reference [17] and show how these can be modeled using our preference approach to allow the caller to express additional preferences and constraints. Again, we contrast the rather cumbersome use of multiple Accept-Contacts in SIP with the use of more intuitive and compact preference expressions.

5.1. Matching feature sets

A matching algorithm for feature sets is given in Reference [17]. The goal of this algorithm is to determine the grade of match between as set of Accept-Contacts and possible SIP contacts to further differentiate between matching contacts.

Tables 2 and 3 give three examples that show, how the SIP matching algorithm basically works: let Ci be the set of SIP contacts to be matched against multiple Accept-Contacts ACj. For each contact Ci the algorithm computes an aggregated score for that contact against each ACj in the contact’s matching set. Let the number of terms for each Accept-Contact ACj be equal to N. Each term in the Accept-Contact predicate represents a single feature tag. If the contact Cj has a term containing the same feature tag, its aggregated score is incremented by 1/N, otherwise the score is not incremented. Based on these

Table 2. SIP feature matching (same as A ⊗ B).

Accept-Contact	AC1	$A \wedge B$	Score
Contact	C1	$A \wedge B$	2/2
Contact	C2	A	1/2
Contact	C3	B	1/2
Matching order		[C1, {C2, C3}]	
Accept-Contact	AC1	$A \wedge B$	Score
Accept-Contact	AC2	A	
Accept-Contact	AC3	B	
Contact	C1	$A \wedge B$	$(1+1+1)/3$
Contact	C2	A	$(1/2+1+0)/3$
Contact	C3	B	$(1/2+0+1)/3$
Matching order		[C1, {C2, C3}]	

Table 3. SIP feature matching (same as A&B).

Accept-Contact	AC1	$A \wedge B$	Score
Accept-Contact	AC2	A	
Contact	C1	$A \wedge B$	$(1+1)/2$
Contact	C2	A	$(1/2+1)/2$
Contact	C3	B	$(1/2+0)/2$
Matching order		[C1, C2, C3]	

rules, the score can rank between 0 and 1. After Accept-Contact/Contact score computation each score is averaged by the number of matched Accept-Contacts for the final score assignment.

Let us for instance consider the simple matching example in the upper part of Table 2 where C1 ($A \wedge B$), C2 (A), and C3 (B) are matched against AC1 ($A \wedge B$). In this case, AC1 is a full match for C1 while it matches C2 and C3 only to 50%. With only one Accept-Contact, no averaging of scores is necessary which yields a matching order of C1 preferred to C2 and C3, both being equally good matches. The same matching order is induced in the second example displayed in the lower part of Table 2. Here the same contacts C1, C2 and C3 are matched against three different Accept-Contacts AC1 ($A \wedge B$), AC2 (A) and AC3 (B).

Interestingly, for both matching examples, we can simply replace the list of SIP Accept-Contacts by one single preference expression $A \otimes B$ that will yield the same matching order [C1, {C2, C3}] (without the necessity of an explicit matching algorithm). Intuitively we can argue that this is due to the semantics of Pareto-accumulation '$\otimes$' which is treating operands a being equally important. Therefore, with AC1 ($A \wedge B$) as the only Accept-Contact in the first example that calls for contacts that support 'A and B', a replacement of AC1 with $A \otimes B$ is justified. In the second example AC2 and AC3 each reassure the call for feature A and B, respectively. This is also covered by the semantics of $A \otimes B$.

A third SIP matching example is displayed in Table 3, where the Accept-Contact stresses a strong preference for A, especially if also B can be provided. Here, matching C1, C2 and C3 with the SIP algorithm against AC1 and AC2 induces the strict matching order [C1, C2, C3]. This is equivalent to replacing AC1 and AC2 with A&B, which again is in tune with the semantics of '&'. The '&' operator treats its first operand as prioritized, which is intuitively the same as reassuring the call for 'A' through an additional explicit Accept-Contact.

5.2. *Numerical preferences*

As a protocol for initiation of multimedia sessions SIP has, very often, to deal with session features that range of integers or floating point values. Examples are the maximization of a video frame rate or the minimization of jitter delay. Numerical values of these features are linearly ordered by standard numerical comparison on integer or floating point sets.

Note that these linear orders are only special cases of the (strict) partial preferences orders that we have dealt with so far and that they can be integrated into partial order preference frameworks naturally. For example Reference [12] defines multiple numerical base preferences that make use of continuous distance functions working on and with numerical comparison. An example is the AROUND preference $P := \mathrm{AROUND}(A, z)$ that aims at feature values in a minimal distance from the value z, i.e. $x <_P y$ iff $\mathrm{distance}(x, z) < \mathrm{distance}(y, z)$, where $\mathrm{distance}(v, z) := \mathrm{abs}(v - z)$. In similar ways numerical preferences to maximize or minimize feature values can be defined. In fact, with numerical ranking any user-supplied numerical scoring function could be used to induce an order on session features.

5.3. *Reject-contacts*

Besides 'positive' preferences expressed through Accept-Contacts that should be met for the fulfillment of a best-possible match, Reference [17] proposes Reject-Contacts to express 'negative' preferences, i.e. explicit dislikes. Using the same syntax as Accept-Contact, a Reject-Contact header explicitly states that a SIP contact should not be contacted if it matches any values of the header field.

With a similar intention, Reference [12] introduces the preference operator NEG(A, NEG-Set$\{a1, \ldots, an\}$) that intuitively states that the desired value for a feature A should not be any from a set of dislikes $\{a1, \ldots, an\}$. Following the above considerations on the replacement of Accept-Contacts through preference expressions we can thus use NEG preferences to replace Reject-Contacts. Using the NEG-operator we can for instance add the preference NEG(video, NEG-Set{wmf}) to our above video session example stating that apart from our previous preference of 'mpeg' over 'h261' and 'quicktime', 'wmf' is the least preferable video codec in our session negotiation and should always be avoided. This means that apart from the explicitly mentioned codecs 'mpeg', 'h261' and 'quicktime' every other possible codec is considered still better than 'wmf'.

5.4. *SIP feature tags: require, explicit*

In addition to the discussed feature sets in SIP Accept- and Reject-Contacts, Reference [17] introduces the feature tags 'require' and 'explicit' for the further classification of feature predicates. Whereas all SIP preferences that we have discussed so far were treated as soft constraints these supplementary feature tags can be used to express somewhat hard matching criteria. The precise behavior depends heavily on whether 'require' and 'explicit' are present alone or in combination.

When only ‘require’ is present, it means that a contact will not be used if it does not match. If it does match—full or only partially—the contact is used. Together with other preferential concepts from Reference [17] (e.g. q-values and feature matching) ‘require’ indicates a preferential match to SIP contacts that is treated as a soft constraint as long as possible. Only in the case where definitely no (partial) match to a required feature is found, the session negotiation will fail. When only ‘explicit’ is present, it means that all potentially applicable contacts are used. However, those that explicitly indicated the marked feature will be preferred in the matching.

Note that we will discuss implicit features as the complement of explicit features in the next subsection. The combination of both, ‘explicit’ and ‘require’, qualifies a feature as being a hard constraint. A feature marked with these two tags must be met, otherwise the matching fails. This is beyond the scope of any preference relaxation framework and matching. Instead, SIP contacts not according to ‘explicit’ and ‘require’ features will be singled out prior to the actual preferential matching.

The exclusive use of ‘require’ has no preference counterpart in Reference [12] since preferences are handled as pure soft constraints only. However, intuitively it could still be obtained by handling preference terms marked with ‘require’ as semi-hard constraints in the matching, i.e. preferential matches are allowed along the lines of an explicit preference, other than that a match will fail. This requires an extension to the preference framework and the matching semantics of Reference [12]. On the other hand accommodating the exclusive use of SIP’s ‘explicit’ in preference expressions is straightforward: modeling explicitly tagged features as prioritized preferences will provide equal semantics.

5.5. *SIP feature tags: implicit*

SIP also offers the use of so-called implicit preferences. Implicit preferences only occur, if no explicit preferences in an Accept-Contract have been given. The basic notion is that if a user explicitly specifies preferences, they should always be respected as top priority. But if no preferences should be given some typical assumptions or stereotypes can help to lead to better quality in service provisioning (cf. step 4 in our running example in Figure 1).

For example the typical notion that users prefer high resolution content over low resolutions would be an appropriate thing to add, if a user did not specify something explicitly. Thus, with respect to the available bandwidth (e.g. the constraints added by network nodes) a user would be generally better served, if—in the case that there is a choice between high and low resolution content—always the high resolution content would be delivered. However, if a user would have given an explicit preference on low resolution content (maybe due to limited capabilities of his/her client device) this preference would have to be respected and no other implicit preference would have been added that could possibly overwrite the user preference. This semantics closely resembles the notion of usage patterns as given by the preference frameworks in Reference [2] and thus, can also be easily incorporated into our preference framework given here.

6. EFFICIENT PREFERENCE PROPAGATION

In addition to an intuitive and compact preference model together with basic evaluation methods, the efficiency of preference-based service management in wireless networks also depends on the efficient use of the available data rates. Especially, the data exchanged on the last (wireless) link has to be reduced to a minimum to allow many users sharing a cell or access node. We assume the data to be handled and stored for describing user preferences to be very large since it applies to various kinds of applications in future, widely exceeding what is described, for example in Reference [18] so far. There are several solutions addressing compression of extended service signaling data. This section explains mechanisms additional to compression strategies to allow more efficient preference propagation by:

- an early reduction of the preference data,
- the provisioning of (default) user preferences in the network and
- an early preference matching for request routing.

6.1. *Preference reduction*

In order to reduce the traffic load imposed by the propagation of a large set of preferences through the network, we employ pre-filtering to the profiles and preferences. In particular, we make use of the fact that in many cases one or more of the desired capabilities simply cannot be granted due to technical reasons. For instance if the network connections cannot provide sufficient support or bandwidth for a certain feature, for example due to a limited available data rate on the last link, we can also immediately remove the preference concerning this feature or at least reduce it by adding certain limiting constraints.

Thus, in the case of network limitations the feature predicate sets will already at an early stage be relaxed along the given preference order. An example for such a preference filtering is given in Figure 1. Due to a limited available data rate on the wireless access link of the current connection the preference set for the video resolution is reduced to 'resolution(low)' alone while the video codec and the language preferences remain untouched.

For efficient preference propagation and filtering we consider the following mechanism.

- We use the above described model to describe the preferences and the network conditions.
- These preferences are normalized, which allows an automatic matching and detection of reduction potentials [12].
- In passing profile information through the network all preferences that become irrelevant due to network conditions can be removed and replaced by appropriate constraints.

In our example, the preference reduction could be done in the base station, or, what would be most efficient to save bandwidth, the terminal being also aware of the current network conditions already reduces the transmitted preference set (cf. step 2 in our running example in Figure 2).

6.2. *Network-hosted preferences*

To further leverage the signaling load imposed by propagating preferences over the last hop of the network, our solution refers to the assumption that not necessarily all preferences need to be stored on every user device. Instead, our system enables users to store common profile information on central network entities and provide minimal (request dependent) overwrites for specialized devices. These default preferences, much like the implicit preferences, could be used in the case when only minimal or no preferences are given in a request (cf. step 4 in our running example in Figure 2).

It is conceivable to classify preferences and preference patterns with the help of ontologies that provide further knowledge about preferences and service requests. Adequate ontologies could allow to add suitable default preferences to the service request preference set, if the semantic relationship between the preferences is understood. In this way preferences are added that for example belong to a certain class of user requests, without the need for the user to explicitly specify these preferences and without imposing extra load on the network (cf. section 5.5).

6.3. *Preference-based request routing*

Preference-based personalization of services may not only address the selection of endpoints servers in the home network, but may also support a selection at an earlier place on the request route (cf. step 3 in our running example in Figure 2). Here, we assume a more general understanding of the communication endpoint address that does not refer to an end user in one home domain, but to the description of content such as a specific video or information about a common subject. In such case, referring to the mobile multimedia framework in Section 2, an end system providing certain information could already be available in the visited network. Preference could help here to specify and to decide whether the request should nevertheless be routed to the home domain. This concept allows early decision making and avoids that all requests have to be routed to home domain where the service is known to exist. However, such routing capabilities, coming more from peer-to-peer systems, are not specified for multimedia signaling systems like SIP yet.

7. RELATED WORK ON PREFERENCE FRAMEWORKS

The introduction of preference-based frameworks for the incorporation of information about specific users in requests goes back to Reference [14]. First systems that relied on the relaxation of constraints if no perfect matches could be found are presented in References [5, 16] for the area of cooperative retrieval. Theoretical properties of these relaxation techniques and operators for building complex preference expressions are in detail addressed in References [4, 12].

Targeted at Internet portals and service catalogues, concepts for preference-based service discovery and service selection are already described in our previous work in References [2, 3]. These papers describe a two-level mechanism for a system-assisted personalized selection of user-centered services using an ontology-based model of the service offerings, individual user preferences and typical usage patterns. In particular, web services were addressed as an example to illustrate the increasing service diversity and need for advanced support for personalization.

Q2 8. CONCLUSION

Q2

In this paper, we have presented a new concept for the network-support of preference-based session management and an advanced solution for preference handling for multimedia systems based on the SIP. In particular, we refer to the concept proposed for SIP [17] to discuss the benefits of our approach. Preference handling such as in Reference [17] could be improved to express user wishes and other soft constraints more naturally to allow better personalization and to reduce the preference handling complexity.

Using a practical example of a multi-lingual multimedia call center, we exemplified how preference matching and service personalization can be achieved in accordance with advanced networking standards, however using more intuitive means. Our solution can be embedded in IP-based mobile communication systems and leverages several profiling features, such as the centralized provisioning of default service preferences. However, our approach is not limited to mobile or IP-based architecture, but applies to any kind of multimedia communication system.

One challenge for preference handling in mobile communication systems is the overhead implied by preference signaling in the network and on the wireless link in particular. Complementing compression solutions, our approach addresses this problem by mechanisms such as allowing preferences to be stored in the network to be updated by preferences coming over the wireless link, or processing of the preferences to delete unrelated data or the concept of implicit preferences.

To conclude, we would like to point out that an efficient network support of personalization not only improves operator service handling, but also serves as a capability to be offered to third party service providers to leverage their personalization efforts and to encourage them to enter the compelling market of personal services.

REFERENCES

1. 3GPP. The third generation partnership project. http://www.3gpp.org.
2. Balke W-T, Wagner M. Towards personalized selection of web services. In *Proceedings of the International World Wide Web Conference (WWW)*, Budapest, Hungary, May 2003.
3. Balke W-T, Wagner M. Cooperative discovery for user-centered web service provisioning. In *Proceedings of the International Conference on Web Services (ICWS2003)*, Las Vegas, Nevada, June 2003.
4. Brewka G. Logic Programming with ordered disjunction. In *Proceedings of the 8th National Conference on Artificial Intelligence (AAAI)*, Edmonton, Canada, July 2002.
5. Chaudhuri S. Generalization and a framework for query modification. In *Proceedings of the International Conference on Data Engineering (ICDE)*, Los Angeles, USA, February 1990.
6. [Q3]Davey BA, Priestley HA. *Introduction to Lattices and Order.* Cambridge Mathematical Textbooks, Cambridge University Press, 1990.
7. EURESCOM Project P1203. The operator's vision on systems beyond 3G. http://www.eurescom.de
8. Guttman E, Perkins C, Veizades J, Day M. Service location protocol, Version 2. IETF RFC 2608, Internet Engineering Task Force, June 1999.
9. Kellerer W. Server architecture for network-independent service control in heterogeneous communication networks. PhD Thesis, Munich University of Technology, Institute of Communication Networks, Munich, Germany, January 2002.
10. Kellerer W, Berndt H. Next generation service session signaling. In *Proceedings of the TINA 2002, Petaling Jaya*, Malaysia, October 2002.
11. Kellerer W, Wagner M, Balke W-T. Preference-based session management for personalized services. In *Proceedings of the MOMUC 2003*, Munich, Germany, October 2003.
12. Kießling W. Foundations of Preferences in database systems. In *Proceedings of the 28th International Conference on Very Large Data Bases (VLDB)*. Hong Kong, China, 2002, pp. 311–322.
13. Klyne G. A syntax for describing media feature sets. IETF RFC 2533, Internet Engineering Task Force, March 1999.
14. Lacroix M, Lavency P. Preferences: putting more knowledge into queries. In *Proceedings of the International Conference on Very Large Databases (VLDB'87)*, Brighton, UK, 1987.
15. [Q4]Naghshineh M. Context-aware computing. *IEEE Wireless Communications Magazine*, Special Issue on Context-Aware Computing October 2002.
16. [Q5]Motro A. VAGUE: a User Interface to Relational Databases that Permits Vague Queries. In *ACM Transactions on Office Information Systems (TOIS)* 6(3), 1988.
17. [Q6]Rosenberg J, Schulzrinne H, Kyzivat P. Caller preferences for the session initiation protocol (SIP). IETF draft, draft-ietf-sip-caller-prefs-10.txt, October 2003. Work in Progress.
18. [Q6]Rosenberg, J, Schulzrinne H, Kyzivat P. Indicating user agent capabilites in the session initiation protocol (SIP). IETF draft, draft-ietf-sip-callee-caps-01.txt, October 2003. (Work in Progress).
19. Rosenberg J, Schulzrinne H, Camarillo G, Johnston A, Peterson J, Sparks R, Handley M, Schooler E. SIP: session initiation protocol. IETF RFC 3261, Internet Engineering Task Force, June 2002.
20. SUN Microsystems (2003). Jini network technology specifications v2.0. http://wwws.sun.com/software/jini/specs/index.html
21. Wagner M, Balke W-T, Hirschfeld R, Kellerer W. A roadmap to advanced personalization of mobile services. In *Proceedings of the International Conference DOA/ODBASE/ CoopIS (Industry Program)*, Irvine, CA, US, 2002.

Q3

Q4

Q5

Q6

Q6

AUTHORS' BIOGRAPHIES

Wolfgang Kellerer heads the service platform & middleware technology group at NTT DoCoMo's European Research labs in Munich, Germany. Before he joined DoCoMo, he worked at the Institute of Communication Networks at Munich University of Technology. In 2001, he was a visiting researcher at the Information System Laboratory of Stanford University, U.S.A. He holds a Ph.D. and a Masters degree in electrical engineering and information technology from the Munich University of Technology.

Matthias Wagner is a senior researcher in the service platform & middleware technology group at NTT DoCoMo's European Research labs in Munich, Germany. Prior to joining DoCoMo, he worked for the Department of Artificial Intelligence at the University of Ulm and the department of Databases and Information Systems at the University of Augsburg. He holds a Master degree in Computer Science from the University of Ulm and a Ph.D. from the University of Augsburg.

Wolf-Tilo Balke is a research fellow in the database group at the Electrical Engineering and Computer Science Department of the Univer-sity of California, Berkeley, U.S.A. Prior to joining UCB, he worked for the chair of Databases and Information Systems at the University of Augsburg. He holds a master degree in Mathematics and a Ph.D. in Computer Science from the University of Augsburg. Since 2002, he is a member of the Emmy-Noether-Program of the German Research Foundation (DFG).

Henning Schulzrinne is associate professor in the Electrical Engineering and Computer Science Department at the University of Columbia, New York City, U.S.A. Prior to that, he was a postdoc at AT&T Bell Labs, Murray Hill, and a researcher at GMD Fokus Berlin. He holds a Masters degree and a Ph.D. in Computer Science from the University of Cincinnati, Ohio, and the University of Massachusetts, Amherst respectively.

Author Query Form (ETT/990)

Special Instructions: Author please write responses to queries directly on Galley proofs and then fax back. Alternatively please list responses in an e-mail.

Q1: Author: Please check short title.

Q2: Author: Please check the heading.

Q3: Author: Please provide publisher location.

Q4: Author: Please provide complete details.

Q5: Author: Please provide publisher name and location.

Q6: Author: Please update.

Q7: Author: Citation of Figure 3 is not in sequence. Please check.