# Generic Conversion of Communication Media for supporting Personal Mobility

**Tom Pfeifer, Radu Popescu-Zeletin**
Technical University of Berlin
Open Communication Systems (OKS)
Franklinstr. 28, 10587 Berlin, Germany
e-mail: pfeifer@fokus.gmd.de

**Abstract:** The Intelligent Personal Communication Support System is introduced as an application for multiple media conversion tools, embedded in a context of personal mobility, service personalization and service interoperability support. After discussing models for conversion in theory, the current conversion technology is evaluated. The necessity of an integrated framework of flexible converters and a generic converter model are derived and automatic management of conversion quality is discussed.

**Keywords:** Media Conversion, Quality of Service, Personal Communication Support, Personal Mobility, Service Personalization, Service Interoperability, TINA Service Architecture.

## 1. Introduction

A lot of effort has been spent in the previous years to build a world full of multimedia systems. We have experienced great progress in creating digital, computerized multimedia data, compressing and decompressing them efficiently, distributing them over innovative high speed networks, and presenting them with best effort quality on dedicated multimedia devices and workstations.

At their workplace, people can now communicate with each other in high quality, employing all means of communication media their environment supports. However, when they leave this well equipped place, they are not willing to be cut off the information stream. Therefore, they carry mobile communication devices, and they find less properly or differently equipped places at home, in a hotel, or at their alternative workplace.

While they now have to tolerate a lower style of presentation of the information, e.g. because they cannot attend the video conference, or cannot receive faxes or electronic mail with their simple telephone, they are still interested in the semantic of this information. This is the point where the conversion of different communication media can be introduced, with the effect of largely increasing flexibility regarding the choice of communication devices and the transported media. [1]

Currently, such media conversion is done as a dedicated stand-alone process. Faxes received by a computer can be converted to text by the OCR software (Optical Character Recognition), built into the fax software. A speech synthesis program (TTS: Text to Speech) might read incoming electronic mail to the recipient. These processes have to be configured manually. It is difficult to choose an appropriate process automatically, and trying to combine such conversions may lead to unpredictable results, because it is vulnerable to error propagation.

Therefore, this paper investigates a generic approach of converting various media into each other, preserving the semantic of the information contained, complete or in parts. It aims to choose converters automatically for a specific problem in communication, involving the controlled combination (concatenation) of various converters. The conversion should work in distributed computing environments [4, 11].

This approach is applied to the concept of *"Personal Communication Support (PCS)"* which provides people with a new dimension in communication. In general, the concept allows users to establish their own personalized communication environment by addressing three important aspects, namely:

- *Personal Mobility*, which denotes the mobility of the user in *fixed* networks and wireless networks, allowing the user to make use of communication capabilities available at different locations, i.e. at any place;
- *Service Personalization*, including personalized call / reachability management allows the user to configure his communication environment and control his reachability according to his specific individual needs, i.e. *if, when, where, for whom* he will be reachable; and
- *Service Interoperability* in distributed multimedia environments, addressing the interoperability between different types of communication services and terminals, allowing users to maximize their reachability. In this context, the multimedia capabilities are required that enable dynamic content handling and conversion of different media types and formats in order to deliver information in any form.

Therefore the GMD Research Center for Open Communication Systems (FOKUS) and the department for Open Communication Systems at the Technical University Berlin have developed on behalf of DeTeBerkom , a subsidiary of the Deutsche Telekom

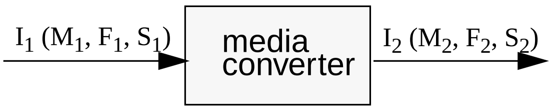



Fig. 1. Media converter system

AG, a TMN-based *Personal Communication Support System (PCSS)* [10] [11], which addresses the first two aspects of PCS, namely personal mobility and service personalization. In order to provide full PCS capabilities within emerging CORBA/TINA based environments, both departments started in 1996 the joint development of an "Intelligent Personal Communication Support System (iPCSS)".

The paper is outlined as follows: Section 2 develops theoretical models for multimedia conversion and introduces the concept of converting media and formats. Section 3 focuses on current converter technology and considers the state of the art of some complex problems. It derives a generic conversion matrix, which quality-of-service management is discussed in Section 4. Then, Section 5 describes the application of conversion in the context of Deutsche Telekom research projects for enhancing Personal Communication Support [18, 19]. After a quick introduction in relevant TINA concepts, chapter 6 describes the iPCSS architecture, its components and their interaction. The remaining sections conclude the paper and provide acronyms and references.

# 2. Multimedia Conversion

## 2.1. Modeling

A media converter may be defined as a system entity, which input is information $I_1$ with the semantic $S_1$, carried by a specific medium $M_1$, using a specific form (or format) $F_1$. We obtain information $I_2$ as output in another Medium $M_2$ in format $F_2$, carrying a semantic $S_2$ (see Figure 1).

The quality of conversion can be measured by comparing the input and output semantic, $S_1$ and $S_2$, which should be preferred to be as close as possible, or having a predefined reduction.

Considering the well-known Shannon's model for information transmission [20], consisting of source and drain, encoding and decoding layers, and the transmission channel, the media conversion could be seen as a specific kind of encoding or decoding, and therefore placed among the encoder/decoder stacks (see Figure 2a).

However, this first approach of modelling fails when the converted information is retransmitted over a different channel after converting, instead of being converted near the sender or recipient. Additionally, the conversion process might require further decoding and encoding of the received and transmitted signal, at least in the physical layer, but often in higher layers as well. Therefore, Figure 2b provides a more general model of conversion. It reminds us of a protocol stack in a network layer model, in particular of the bridge or gateway functionality. Indeed, the conversion process could be considered as bridging two different worlds of communication.

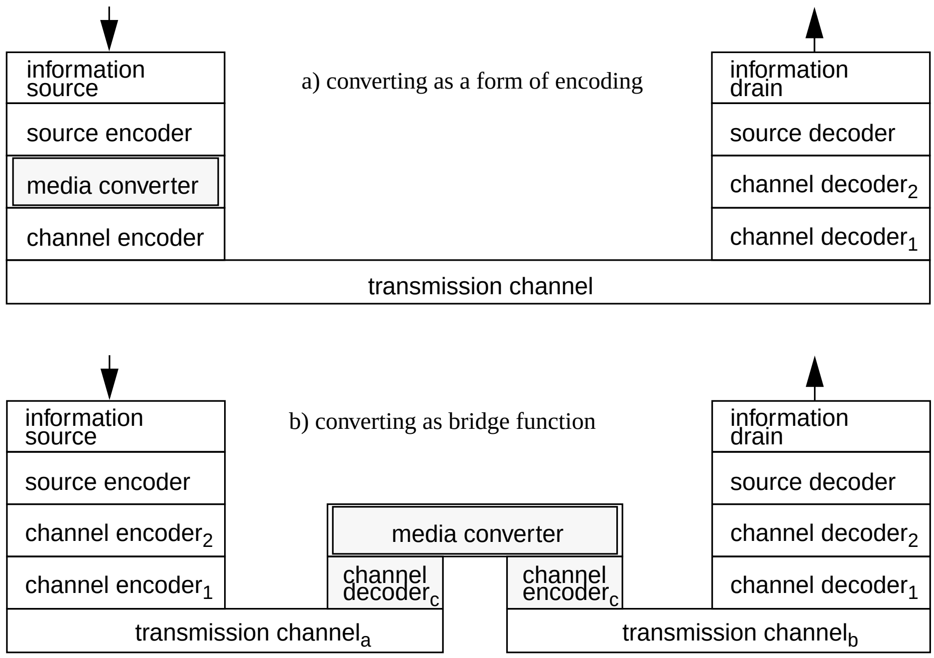
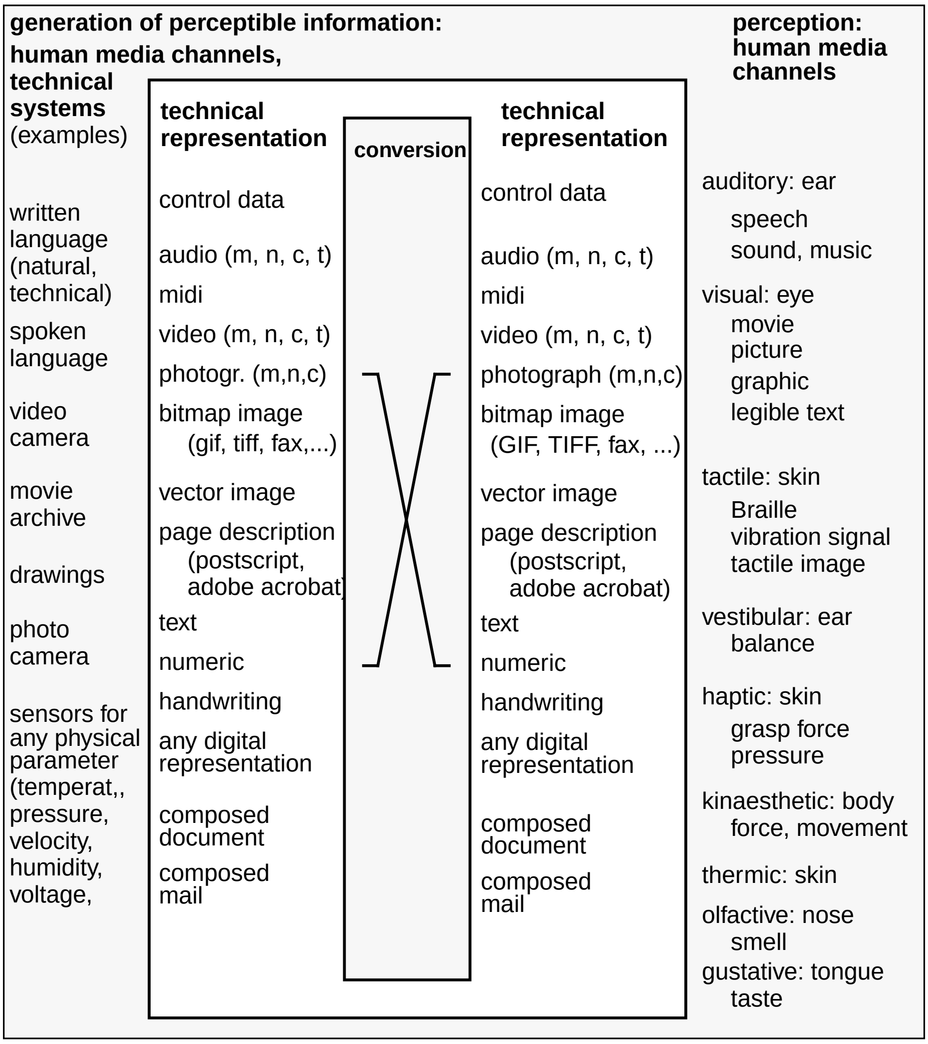



Fig. 2. Modelling information transmission and media conversion

.




parameters:
m, n: media dependent parameters
(frame/sampling rate, quantization, resolution, size, color depth, etc.)
c: applied compression technique
t: time, duration, etc.

Fig. 3. Generic conversion matrix

For specifying the model in more detail it is now necessary to consider the kinds of intended conversions. For this purpose, we need to consider that all information is intended to be perceived by human beings. However, this information could be generated by other humans as well as technical systems.

Figure 3 lists on the left hand side some example sources of information. In the inner block, the technical representations of these information sources are specified. A technical converter would now change one technical representation into another, trying to leave the semantic intact, or to produce a dedicated subset of the input semantic. The technical representation at the output of the conversion can than be applied to the human senses, serving as perception channels for the information.

### 2.2. Media Type Conversion vs. Media Format Conversion

As we see in Figure 3, even the same class of technical representation can have various parameters, some specific to a certain medium (e.g., colour depth of a picture), some more general (time parameters). Obviously, modifying important parameters of the same medium can be considered as a conversion too.

Therefore, we can distinguish two major classes of conversion:

- Media type conversion, and
- Media format conversion.

The latter class of conversion does not alter the medium type. It converts into another format within the same type, or modifies the medium within the same format (e.g., scaling, colour reduction, etc.). A medium format converter is expected to be highly flexible, allowing most possible conversion combinations, accepting as well as delivering the formats with maximal parameterization (size, sampling rate, etc.).

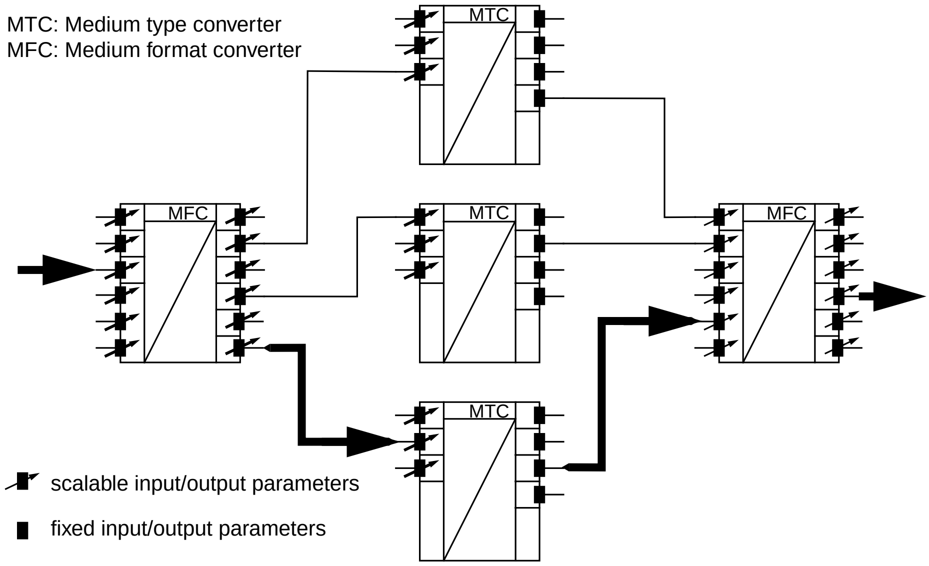



Fig. 4. Medium type conversion with format adaptation

This first class of conversion alters the type of the medium. As we will see later, the converter is in most cases specialized on a specific task, because it is highly complex (like TTS, OCR, speech recognition, etc.). The converter is expected to be very limited in acceptance of input formats and generation of output formats. Implementing multiple converters, e.g. from different manufactures, in a generic environment, leads to the necessity to accompany the type converter with two format converters at the input and the output, respectively, in order to achieve a larger range of possible formats.

This concept leads to a rudimentary framework of type and format converters, as drafted in Figure 4, where multiple, different implementations of a type converter for a specific task of conversion are accompanied by an input and an output format converter. Together, they form the smallest structure in a converter framework. Note that in case of concatenating two or more type converters (MTC), i.e. two or more structures like Figure 4, the intermediate format converter (MFC) can be utilized by both type converters commonly.

The figure demonstrates the availability of three different type converters, delivering and accepting different input/output formats. For a specific task of conversion, a conversion path has been chosen (fat arrows) which is most appropriate to the current resources and the desires of the user.

Having implemented a working universal conversion matrix as depicted in Figure 3 would enable the system to convert any possible medium into any other. While this would be academically demanding, practical considerations will lead to some discretions that are required within a given environment. These are discussed in the next section.

## 3. Conversion Technology

### 3.1. Usability

While the previous section approached the theoretical modelling of media conversion, this one should discuss the technology available, and consider practical constraints.

Table 1 presents a collection of practical examples for illustration, focusing on requirements in multimedia communication and information systems, with the attitude that even conversions that sound strange in the first place might be of very practical relevance. E.g. reading temperature values over a phone line is a conversion of temperature into audio, or converting audio into a vector graphic may visualize the structure of a recorded message.

The chaining of converters is illustrated for the temperature example, which has been implemented for demonstration purposes (Figure 5).

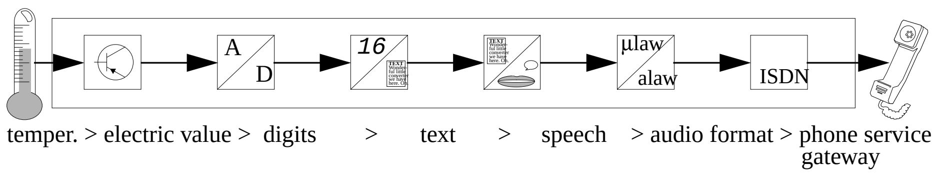



Fig. 5. Converter chain for temperature to speech conversion with telephone delivery

The range of conversions varies tremendously in effort, cost and required resources. Some kinds are easy to implement with two lines of C code without any patent or license requirements (like converting one audio sample into another), while others are highly complex, requiring sophisticated solutions that might be available only as commercial products, and may be covered by patent or copyright protections (e.g., TTS). Some conversions are purely algorithmic, while other require approaches of artificial intelligence (e.g., speech recognition), or require pipelined processes of decoding, editing, and re-encoding.

| source | drain | process | example / application |
|---|---|---|---|
| text | text | format conversions | ASC II 7-bit -> 8-bit |
| bitmap image | bitmap image | | tiff ->JPEG |
| video | video | | MPEG -> H.261 |
| audio | audio | | μ-law -> a-law |
| bitmap image | vector image | vectorization | |
| audio | vector image | visualization | length and dynamic of a message |
| text | speech | speech synthesis | TTS |
| speech (audio) | text | speech analysis & recognition | commands, dictation |
| bitmap image | text | OCR | OCR |
| fax bitmap | speech | OCR+TTS | fax reading |
| temperature | bitmap image | temperature distribution of objects | weather map, medical map |
| temperature | audio | temp -> text -> TS: value reading | weather report |
| numerics | image | visualization of statistics | charts |
| text | tactile information | feed Braille output device | blind reading |
| control data | tactile information | vibration device | pager signalling |
| audio | control data | speaker recognition | prioritizing, authorization |
| photograph or video | text | face recognition, mimic recognition | (e.g. very low bitrate compression) |

Table 1 Selected examples of media conversion

A generic intermediate format is often used to reduce the number of required tools in the conversion process. The latter may be of the same kind than the converted medium (e.g. image format conversion), or a different one (e.g. fax > [ocr] > text > [speech synthesis] > reading). In order to avoid losses in quality the intermediate format temporarily needs the resources of the highest possible quality among the involved formats. E.g., in converting bitmap images, the intermediate format needs resources for the largest possible pixel resolution and true-color depth per pixel.

The advantage of requiring less tools is paid with

- the necessity of more resources (e.g. for an intermediate format or for computing time),
- the necessity to convert twice, causing a delay and – in some cases – a reduction in quality.

A possible solution for this problem is a hybrid approach, i.e. to use dedicated tools if available, and intermediate formats in other cases. At this point, it becomes necessary to classify the demands of conversion:

1. highest priority, very often used, maximum speed required:
   This category should be implemented as a dedicated service, featuring a dedicated software solution including supplemental hardware, for performing a one-step conversion.
2. medium priority, often used:
   In the intermediate area compromises are possible, dedicated software would be an option, one-step conversion recommended.

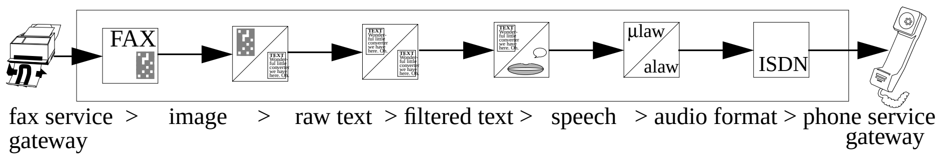



Fig. 6. Converter chain: fax reception, conversion to text and speech, telephone delivery

3. lower priority, rarely used:
   For services that are seldom employed, the conversion via a generic intermediate format is possible. The resulting multiple-step conversion requires a preliminary definition (or dynamic configuration) of a conversion path, e.g. instructions which converters and intermediate formats are appropriate to achieve the maximum quality or shortest conversion time, respectively.

Even when the same pair of media is converted, different performance, quality and gradability might be required – depending on the application. E.g., speech recognition (speech –> text) can be focused on dedicated speakers for dictation purposes, or on a very limited vocabulary produced by unknown speakers for command or keyword recognition.

### 3.2. State of the Art in media type conversion

For media conversion, the most complex problems of TTS, OCR and speech recognition should be shortly discussed. All three of them have strong relations to Natural Language Processing/Representation and Artificial Intelligence.

- *Text-to-speech*

Speech synthesis is the task of transforming written input to spoken output. The input can either be provided in a graphemic/orthographic or a phonemic script, depending on its source. Two major classes of algorithms are the concatenation of previously recorded or generated utterances, or the synthesis of waveforms according to a model of the human voice tract. [22, 23]

In the former case, the easiest way is to just record the voice of a person speaking the desired phrases. This is useful if only a restricted number of phrased has to be produced, like pre-specified messages. More sophisticated are algorithms which split the speech into smaller pieces, down to phonemes or their duplications, diphones, as well as syllables. Most commercial TTS systems employ these methods.

The latter case requires format synthesis, which is done by digital signal processing. This version is in more experimental states.

While an intelligible speech can be synthesised today, it sounds still rather artificial and monotone. Therefore, research goes on to apply and improve the prosody of the speech, that is the intonation and phrase melody. While it can partly be done by analysing the grammatical structure of a sentence, it is difficult to find out which words have to be stressed for their relevance to the meaning of the statement. This problem can only be solved applying background knowledge and artificial intelligence. [24] This technique is even more important for the other speech related conversion: automatic speech recognition.

Most manufacturers offer text reading and e-mail vocalization as one of their primary applications. However, in most cases the service is not generic but proprietary, tailored to a dedicated environment of operating system and hardware platform. Mostly, audio is sent to the speaker line without the possibility of further processing, audio format conversion and forwarding.

- *OCR*

Optical Character Recognition identifies valid text blocks in an image, and maps the image representation of the symbols to their ASCII equivalents. Remaining ambiguities in recognition may than be processed by spelling checkers and language processors.

Most systems are designed for recognition of printed characters. Recognition of hand-written language experiences similar problems as speech recognition in recognizing boundaries and requiring more sophisticated interpreting systems.

Many commercial systems perform 99.9% correctness for clean text images in a small number of simple fonts. It is difficult to deal with font and scale variations (omni-font), as well as with noise (from digitizing and intensity variations) [21], particularly when the input comes out of a telefax system, or when included graphics have to be considered.

The recognition of typographical pattern written by hand is a specific sub-class of OCR. It is much more complex due to the variation of people personal handwriting styles, and the wide unlikeness between the writing style of different people.

Therefore, similar to speech recognition discussed below, two major trends lead to useful applications. Firstly, the system could be trained on a specific writer. This approach is very useful for personal or personalized devices, such as handheld computers (e.g. Apple Newton). Secondly, the set of legal characters for recognition could be limited (similar to command recognition in speech). This approach is widely used e.g. in sorting letters automatically in snail-mail distribution centres.

- *Speech recognition*

Automatic speech recognition is the process by which a computer maps an acoustic speech signal to text. Automatic speech understanding is the process by which a computer maps an acoustic speech signal to some form of abstract meaning of the speech.

Such systems can be speaker dependent, adaptive, or independent. The first kind is developed to operate for a single speaker, therefore usually easier to develop, cheaper to buy and more accurate, but not as flexible as speaker adaptive or speaker independent systems. A speaker independent system is developed to operate for any speaker of a particular type (e.g. American English). These systems are the most difficult to develop and most expensive to buy and their accuracy is lower than speaker dependent systems. However, they are more flexible. [25, 27]

The size of vocabulary of a speech recognition system affects the complexity, processing requirements and the accuracy of the system. Some applications only require a few words (e.g. numbers or commands only), others require very large dictionaries (e.g. dictation machines).

Another problem relates to continuous speech vs. isolated-words. In the latter case, a pause between saying of each word makes the recognition of word boundaries easier. In continuous speech, there are much less hints for these boundaries, additionally, neighboured words influence each other (coarticulation).

The process of speech recognition starts with the digital sampling of speech, followed by acoustic signal processing, mostly including spectral analysis. Next, phonemes, groups of phonemes and words have to be recognised. Systems based on Hidden Markov [26] modelling are mostly used.

### 3.3. State of the Art in media format conversion

For converting one format of the same medium into another, tools exist for many platforms for text, bitmap images and audio. They are mostly in the public domain, and perform well as software solutions [32]. Converting video formats requires the appropriate encoding/decoding hardware and software for the compression methods involved [28, 29, 30, 31].

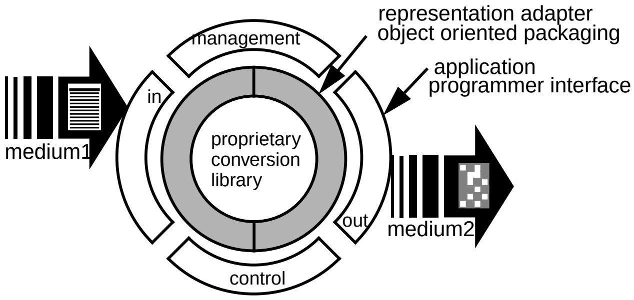



Fig. 7. Generic converter model

### 3.4. Generic Converter Model and Generic Converter Framework

Like outlined above, a *generic converter model* has been designed. A generic converter framework has been implemented according to the converter model in order to abstract from the requirements of the underlying converter software.

In the generic converter model shown in Figure 7, the conversion is executed in a core library, which is proprietary to the manufacturer. A representation adapter layer covers the specific properties of the core library and unifies the format requirements for the conversion framework. For extrinsic converter solutions not complying completely with the object oriented design of the framework, the representation adapter also realizes the object oriented behaviour.

The outer layer delivers management and control interfaces. Input and output adapters modify the behaviour of the converter to specific requirements of the medium type and might perform some pre- or post-processing.

In order to keep track with personal communication related research activities and the corresponding platform developments, the iPCSS will migrate towards conformity with the Telecommunications Information Networking Architecture (TINA), developed by a consortium which is currently in the focus of worldwide attention [13]. This architecture is based on distributed computing (e.g. based on CORBA [16]), objectorientation, and other concepts and standards from telecommunications and computing industries.

The details of the converter model are specified in this project in a way that makes the migration towards TINA as easy as possible. [10]

In a framework, particularly in a distributed environment [16], various tools of conversion can be provided for different requirements, with different performance, and at different cost. To form such a framework, the conversion tools utilized must have unified interfaces, the *Generic Converter Interface* (GCI) for various purposes.

The GCI defines initialization, the localization of an appropriate converter, access to the description of conversion quality (see next section), data flow control, management, and finally termination of an instance of a single converter.

The *generic converter framework* has to abstract from the underlying hard- or software used for the actual conversion. E.g. for first testing scenarios, widely used software packages for simple format conversion and complex solutions for OCR, TTS and FAX have to be included by simplifying the access to these packages as much as possible.

The main purpose of the converter framework is to identify the general behaviour of filter and converter software packages and to define a unique subset of functions which can control, start, stop, manage and configure these software packages. Furthermore, general methods for the deliverance and transport of the conversion data have to be identified.

Currently used software packages can read and write there data from files, pipes or even network-wide sockets. The implementation of the converter framework realizes the GCI and hides the specific maintenance and data transport functions from calling objects like SAPs. Additionally it maintains the QoS matrix (cf. section 4.) of all available QoS parameters for all software packages.

The converter framework initiates and reserves conversion services, validates the state of the started hard- or software, controls jobs and processes as well as the data flow via file redirection, pipes or TCP/IP-based services and database functions for the specification of available soft- and hardware modules used for the conversion.

## 4. Quality of conversion services

As it was stated above, conversions are of different quality due to the nature of the processes involved, and due to limitations in current technology. The conversions might be lossy (like or because of lossy compression), quality-limiting (like conversion to lower resolution), or error-prone (like OCR or speech recognition). When such conversions are applied, these quality problems have to be considered. [33]

### 4.1. Quality control and management

Each conversion consumes resources of the environment and influences the quality of the final output. When concatenated, the influence of each conversion has to be considered in the context of the whole chain of converters. Objective criteria are required for choosing the optimal conversion path among various possibilities for a specific task. However, different tasks can emphasize different parameters for the selection. E.g., synchronous communication requires a short over-all delay, while asynchronous conversion (for later delivery) might emphasize minimal cost or best intelligibility of the output.

### 4.2. Quality-of-Service definition

Similar to the Quality of Service (QoS) parameters in network communication, a lot of parameters could be defined for conversion, encompassing:

- *Intelligibility*
  This parameter is the most important determiner for the correct transport of the semantic of the information during the conversion process. Defined in Webster's dictionary as "capability of being understood or comprehended" or "apprehensibility by the intellect", it describes whether the human perceiving the output of the decoding process is able to recognise its semantic correctly. The term is mostly used in the context of complex conversion, as TTS, OCR and speech recognition.
- *Error probability*
  This parameter is the more technical version of the previous. It describes the probability of bit errors both in conversion and in required transmission between conversions. Both parameters, the intelligibility and the error probability need to be considered in their interference.
- *Quality degradation* due to *lossy compression/decompression*
  This parameter should prohibit multiple lossy compression and decompression processes, if possible (e.g., an image should be compressed into JPEG [31] format only finally, not as an intermediate step).
- *Quality degradation* due to *entropy reduction*
  This parameter describes the loss in quality (and therefore, finally, in intelligibility) when the semantic of the information is partly reduced (e.g. because a 24 bit colour image has to be displayed on an 8 bit screen, i.e. colour reduction, quantification, scaling).
- *Delay*
  Depending on the kind of conversion, there are a start-up delay for a continuous stream of data, delay caused by buffering processes (e.g. an image might be buffered completely before conversion, or a TTS software might wait for the end of a sentence or paragraph before determining the correct prosody), and transmission delay. Such irregularities in converters can produce **jitter** to an considerable extent.
- input/output *data volume and data rate*
  This parameter describes the required storage and transmission resources.
- *Computational resources*
  This parameter determines the requirements of the conversion process regarding hardware platform and computational performance. It has to be considered that a single conversion might be possible on a specific service computer, e.g. speech synthesis, while for multiple parallel conversions specific hardware (like a board of parallel signal processors) might be required.
- *Cost* (respecting tariffs)
  This parameter refers to the computational resources above as well as transmission cost, but is calculated for a single conversion process.

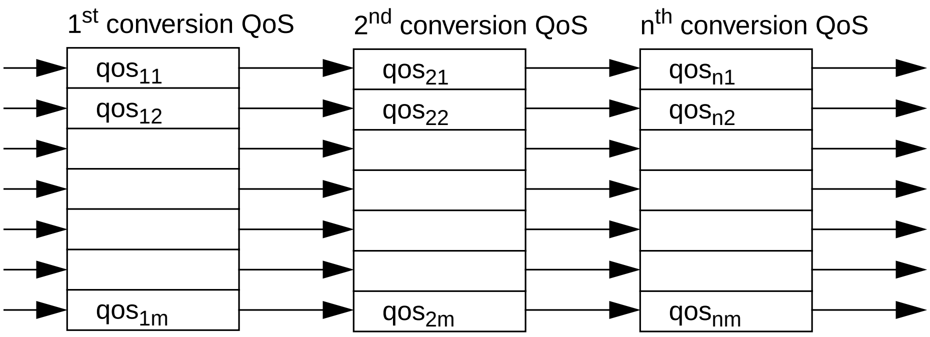



Fig. 8. Propagation of QoS-parameters in concatenated conversions

Comparing different possibilities of concatenating converters for a specific task requires a complex evaluation of the quality parameters involved, performed at runtime. Figure 8 illustrates the propagation of the parameters in concatenated conversions.

For each parameter, the dedicated model for concatenation has to be derived. While delays probably add up, quality degradation and error propagation require more sophisticated calculations, like the evaluation of the sum of squared differences, etc.

### 4.3. Media dependence of QoS parameters

Depending on the existence and the character of losses in the conversion, the process can be reversible or not. E.g., bitmap image format conversion is bitwise reversible as long as color space and resolution are not touched. On the other hand, even printing the output of OCR back on paper will suffer from font modification and formatting losses.

Therefore, this sections emphasizes the difficulties to compare QoS parameters of completely different media due to their heterogenous characteristics. For example,

- *Text* may suffer from character reduction, loss of layout formatting,
- *Images* can be reduced in color space, size and compressed lossy,
- *Audio* can be reduced in sample size, sample rate, suffers from compression loss, and is specifically sensitive to jitter.
- For *Video*, the huge amount of raw data requires compression (complex but effective) in most cases. The standardization efforts have led to a few platform independent formats, however, the required re-encoding in a conversion causes a considerable delay. The problems of still images apply here also, and the unique parameter framerate influences the acceptance by humans severely.

### 4.4. Evaluation and handling of QoS

Algorithms for evaluating various choices of concatenated conversions for the purpose of automatic configuration are currently under development. Most promising are backtracking strategies for analyzing the multiple possibilities, and fuzzy logic comparison of the heterogenous QoS parameters.

The latter method has the advantage that no hard limits have to be defined for individual parameters. Instead, weak parameters can be compensated by other advantages of a chain configuration. E.g., a slightly too long delay might be accepted if the service is cheap enough.

In a first approach, a selection of four important QoS parameters has been made, which cover most considered aspects above. These parameters are:

- *Intelligibility,* covering problems of media synthesis, error probabilities, compression losses, channel noise, and semantic reductions, given in percent (0...100%).
- *Bandwidth*, covering data volume and bitrate, given in bit/s,
- *Delay*, given in seconds
- *Cost*, covering all aspects of tariffing and resource consumtion (transmission and computation), given in countable units.

The narrowest bandwidth of all used components limits the bandwidth of a converter chain. Delay and cost add up and must not exceed predefined limits (depending on the importance of the message). Intelligibility percentages of concatenated components can be multiplied, so that the result is smaller than each value. More sophisticated evaluation models are subject of current research.

Furthermore, the iPCSS is controlled by its users. Consequently, measures taken to cope with the quality problem have to be under their control as well. Such measures could include:

- *Carbon copies:*
  The system keeps copies of the original message, even if it is converted to another medium or another format, and forwarded to another location. For example, faxes that are converted to text (OCR) are kept as the original bitmap for later proving.
- *Limitation of concatenation:*
  Knowing the artifacts of the conversion tool involved, the user might want to limit concatenated conversion steps. E.g., if the OCR produces text results that are only legible with background knowledge, then sending them through a TTS system would make things worse and produce an incomprehensible output. In such a case, concatenation would be disabled after the OCR. Additionally, recursive or reverse conversion has to be avoided.
- *Media Selection:*
  The user might like to decide which transport or displaying media type he prefers in order to maintain a desired quality (for example costs). A possible selection can range from transport, format or media types the user does not like to be included in the conversion process, as well as a description of the resulting format and media type conversion.

## 5. Intelligent Personal Communication Support

The vision for future communication is labelled by the slogan *“information any time, any place, in any form”*. This vision is based on the society’s increasing demand for “universal connectivity” and the technological progress in the areas of mobile computing and telecommunication.

In the remaining sections, the application of automatically configured conversion technology for Personal Communication Support is discussed.

As outlined in the introduction, the iPCSS (Intelligent Personal Communication Support System) aims for the provision of full PCS capabilities. The definition of the PCS concept is strongly influenced by recent research activities in the field of advanced telecommunication and distributed computer sciences, such as IN, UPT, Telecommunication Management Network (TMN) [8], TINA, computer telephony integration [2], mobile/ubiquitous computing [3], and Electronic Location Systems [9].

The trends toward PCS can be viewed in terms of three major areas of research:

- *Mobility support in fixed and wireless networks,*
  enabled by means of *terminal mobility* and *user mobility. Terminal Mobility* by wireless network interfaces and protocols (i.e. cordless, cellular and satellite) is fundamental for the provision of ubiquitous, global connectivity. The next step, *User Mobility* (also called “Personal Mobility” or “Service Mobility”,) will enhance global service access, allowing people to make use of any kind of terminal located at their whereabouts for obtaining access to their services. With this type of mobility support, the user is directly addressed by the means of a personal number instead of addressing the terminal at his guessed current location.

- *Personalization of communication services access and delivery,*
  enabling the user to define his own environment and service working conditions in accordance with his own needs and preferences, with respect to parameters, such as time, space, medium, cost, integrity, security, quality, accessibility and privacy. These parameters are usually stored in a user profile, defining all services to which the user has access, the way in which service features are used, etc. A different aspect of personalization is concerned with the individual management of the user’s reachability (when, where, for whom, by what media), i.e. negative/positive communication filtering.

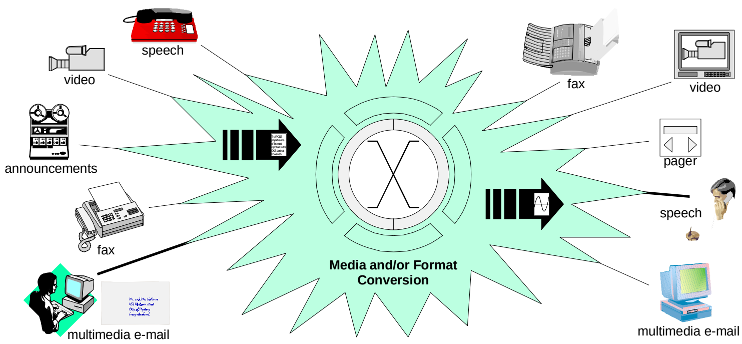



Fig. 9. Priorized media conversion in the iPCSS

- *Dynamic adaptability, service interoperability and flexible media handling,*
  which is a a much less common aspect of personal communication. *Dynamic adaptivness* means that the communication environment is forced to adjust to the user’s needs, knowledge, and preferences as well as to the constraints arising from the user’s current local environment. The general intend is that the system adapts to the user and not vice versa. In particular, this adaptation includes flexible media handling in form of possible *conversion* of one medium into another one (cf. Figure 9).

Consequently, the following design criteria for a system/platform providing PCS capabilities can be derived:

- The addressing of users has to be decoupled and made independent from service, network and terminal capabilities. This leads to the *introduction of personal names/ numbers* for achieving real person-oriented communication.
- The user's service specific (access) control data distributed across multiple communication systems and maintained in service specific data structures has to be unified and integrated into a *common user service data structure* (e.g. a generic *"user profile"* or *"personal call logic"*), configured by powerful *customer (profile) management capabilities*. This includes advanced user registration capabilities for personal mobility support, allowing manual and automatic registration at *locations*, such as offices, instead of specific terminals.
- Furthermore, the vision of delivering information in any form requires the introduction of additional concepts, allowing the *dynamic selection of terminal equipment* at a registered location in accord to the incoming service requirements and/or the called user's preferences. In case of inadequate terminal capabilities *service interworking* and/or *media conversion* require the provision of generic service gateways and/or media converters, as discussed in the previous sections.

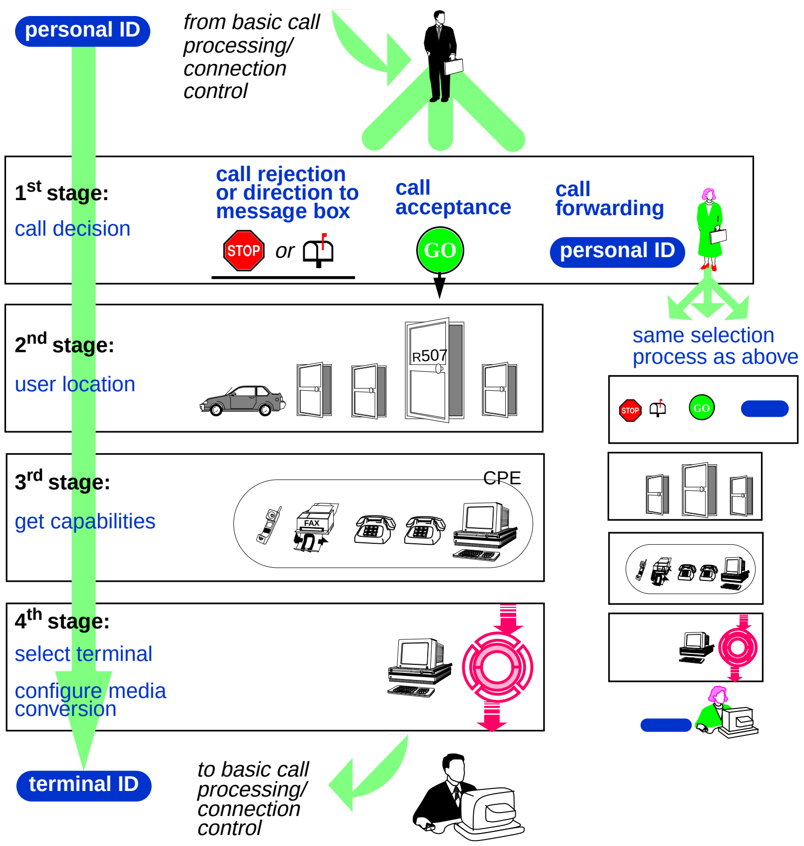



Fig. 10. PCS-based Intelligent Call Processing

Taking all these issues into account the concept of PCS aims for person-oriented and location-oriented communication. Thus PCS enhances the reachability of persons on the one hand while providing reachability control on an individual basis on the other hand. Note that the PCS capabilities are generic, i.e. these capabilities should be provided by an open service / communication platform via a generic application programming interface to many communication services in a uniform way.

In order to illustrate the benefits of the PCS concept, Figure 10 depicts a simplified intelligent call processing model to be performed by an advanced Personal Communication platform. It is characterized by a *four-stage mapping process* that translates a logical user name used as the called party address (i.e. a personal ID) into an appropriate network address (i.e. a terminal ID). This temporary physical address is passed back to the requesting communication service. The mapping process looks as follows:

- *1st*, the evaluation of a user's "Personal Call Logic" provides the *control of his reachability.* The result may be a forwarding to another user, a call rejection, a call redirection to an asynchronous service, e.g. an answering machine, or an acception.
- *2nd*, the exact recipient of the communication invitation has been settled and no further call management will be performed. A *mapping of the user to his location* is made based on user registration data.
- *3rd*, it *maps a location to a virtual communication endpoint* corresponding to a terminal group representing the set of all access devices in the user's current vicinity. An object-oriented modelling of virtual communication endpoints encompasses the knowledge on terminal capabilities, supported services, and selection mechanisms.
- *4th*, an *appropriate terminal ID* from the group of devices is selected and parameterized by a service type, used communication media, and optionally by user preferences. Within this stage, two cases can be distinguished:
  a) In case there exist at least one device of the virtual communication end-point supporting the desired medium of the call, the most appropriate device is selected.
  b) In case no device for the desired medium can be found, further rules of the Personal Call Logic determine whether a *conversion into another medium* is allowed/restricted. Then, the necessary converters are configured and a now appropriate device is selected.

# 6. The iPCSS Architecture – compliant to the TINA Service Architecture

It is beyond the scope of this paper to introduce the whole TINA concept behind the iPCSS. However, some terms should be introduced for common understanding.

## 6.1. Basic TINA concepts of an Access Session

The overall TINA architecture is divided into several architectural aspects [13]. However, within this paper we will focus on the TINA Service Architecture [14]. The latter defines a set of concepts and principles for the uniform design, specification, implementation, and management of telecommunication services and their components. It provides the universal platform for a variety of services in a multi-provider environment. This section concentrates on the most important aspects of the Service Architecture, namely the *"session concept"*, which separates different aspects like service access, service provision and service communication needs, and the generic *“service components”* identified within the service architecture. It has to be stressed, that this description is not complete and highlights only the basic concepts in order to understand the iPCSS design.

Since TINA is intended to support even complex multimedia, multi-party telecommunication services, TINA introduced the notion of a “session”, replacing the traditional notion of a “call”. It provides a means for grouping specific activities in a service during a specific period of time. Three basic types of sessions have been defined [12] [14]:

- The *Access Session* supports an user in accessing, requesting and retrieving telecommunication or information services. The Access Session is service independent and dedicated to a user.
- A *Service Session* represents the core functionality of a service and is therefore service specific. It provides a user or a group of users with an environment to support the execution of a service.
- A *Communication Session* provides an abstract view of connection related resources and supports the activities needed to establish the communication channels (i.e. streams) that may be required between end user systems (and the service provider systems).

TINA services are described in terms of interacting components in distributed processing environments, i.e. *Computational Objects (COs)*. In accord with the session types outlined above, the TINA service architecture introduces a set of generic components for the realization of telecommunication services. This means that each session type is realized by a specific set of interacting COs. In the following we address the main COs defined within the Access Session, which are service independent, and some generic Service Session COs. Note that all these COs can be considered as the generic “construction kit” for the realization of TINA-based telecommunication services.

The generic COs of a TINA Access Session are shortly introduced here, used within both, the user’s own end-system domain and the provider domain.

The customer’s end-system is represented by two COs:

- The *User Application (UAP)* models the specific service application in the user system, i.e. it offers the application’s user interface. It provides the user with access to the Service Session. The latter is modelled by a Service Session Manager (SSM) CO and multiple User (Service) Session Manager (USM) COs, created dynamically by a Service Factory (SF) CO within the provider domain.
- The service independent *Generic Session End-Point (GSEP)* models the minimal set of capabilities required for interacting with the User Agent (see next) to perform service session control and invitation delivery.

Within the provider’s domain, the following COs relate to the Access Session:

- The *User Agent (UA)* represents a user in the service provider domain. It is the contact point of control for personalized session creation, suspension, resumption and deletion.
- The *Personal Profile (PPrf)* maintains the user related constraints and preferences on service access and session execution. It determines the environment, in which the service will be executed for the user.

- The *Usage Context (UCxt)* maintains the knowledge on a pool of resources available to the user for the execution of services. It contains registrations at user terminals and terminals at network access points. For personal mobility support, the UCxt keeps track of the terminals and access points available to the user.
- The *Terminal Equipment Agent (TE-A)* represents a terminal of a user system within the service provider domain. It maintains the capabilities and the state of a terminal from the providers perspective.
- The *Subscription Agent (SubAgt)* is a contact point for accessing subscription information according to users. It interacts with other management related computational objects beyond the scope of thispaper.

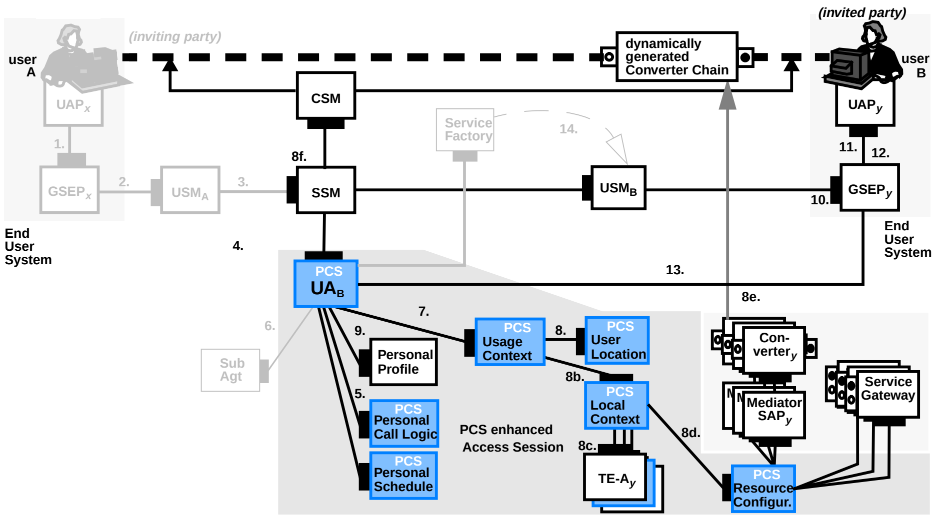



Fig. 11. Components of the PCS-enhanced TINA Access Session

## 6.2. The iPCSS Architecture

The iPCSS architecture is strongly based on the TINA Service Architecture. In general, the capabilities of the iPCSS aim for an enhancement of the TINA Access Session in order to increase the nomadic user's reachability by introducing location-based user registration and coincidentally the dynamic selection of terminals at a registered location. In particular the iPCSS allows the adaptation of information flows to a user's terminal capabilities. Therefore, the iPCSS architecture, which could be considered as an enhanced TINA Service Architecture, defines a set of new COs, mostly related to the Access Session, in order to achieve the intended functionality. In the following we look more detailed at these COs and their interactions.

## 6.3. iPCSS Components

The Access Session enhanced by iPCSS components is given in Figure 11. The following COs have been redefined or newly defined:

*User Agent*

The *PCS-enhanced User Agent (PCS-UA)* is a specialization of the TINA UA defined in the TINA Service Architecture [13]. The PCS-UA has been designed according to the following requirements which are in line with the TINA Service Component Specifications [15]. It

- controls *incoming calls / invitations to join a session*, i.e. the ability to access a user (i.e. alert, instantiate an USM on behalf of the user) and thus protects the user from unwanted communication attempts (cf. Personal Call Logic). Therefore it can be customized (personalized);
- provides (personal) mobility support to locate and reach a user who may move (cf. usage context, user location, local context);
- provides enhanced functionality for inter PCS-UA communication, e. g. for advanced call forwarding which delegates a service invitation to a PCS-UA of the party it is forwarded to.

*Personal Call Logic*

The *Personal Call Logic* of a user determines, how a service invitation should be handled by the invited party. It is the main component to optimize or limit a users reachability and to personalize incoming call handling.

Firstly, the PCS call logic component decides on behalf of the user if an invitation should be accepted or not. Therefore, it contains a rule based system defining the user's intends how to handle incoming calls (e.g., accept, reject, forward, call screening, etc.). A user may define his personal call handling, such as accepting/blocking/forwarding service invitations with respect to the caller, the calling time, the service related to the call, etc.

Secondly, the rules determine whether conversions of communication media are allowed if necessary, enforced in a specific situation, or forbidden in certain cases. These pre-evaluated rules are forwarded to the Usage Context for final selection of terminal devices.

*User Location*

Regarding registration of users, the TINA Service Architecture is designed to handle terminal registrations with the usage context computational object. The more generic PCS approach supports the registration at locations. That moves the terminal selection to the invitation phase. This concept offers the possibility to realize a flexible call handling, that may take current service requirements and network and terminal states into consideration. The possibility of location-oriented user registrations will be achieved by the new *User Location* CO, that supports the PCS-UA by maintaining location information of a mobile user in form of location identifiers.

*Personal Schedule*

This object serves for user registrations triggered by a personal schedule or diary. The registrations may be terminal registrations or registrations at locations (cf. User Location). In addition, the Personal Schedule may contain data and logic for time-dependent service invocations on behalf of the user.

*Local Context*

Many of the information to be handled by PCS capabilities are location related. At the current stage, TINA does not deal with location information yet. Therefore, a new component named *Local Context (LCxt)* has been introduced. Like the Usage Context, the LCxt will maintain knowledge on a pool of resources. *Note that the LCxt is not a user specific component, since it models common infrastructure!* In a service invitation scenario, the LCxt will enhance the access session functionalities to resolve a location identifier of a user to an appropriate network access point and terminal ID. Thus, the LCxt has to contain associations between a specific location (e.g. a room or zone) and terminals (in TINA represented by the TE-A).

From the view of the PCS concept, the Local Context models a virtual communication endpoint. The LCxt is designed to support the Access Session implementing the selection of a terminal at its network access point at the user's current location. Potential client objects of the LCxt are the Usage Context and User Location. At this projects stage, the Usage Context is assumed to query the User Location and subsequently the LCxt pertaining to the user's current location to get terminal information appropriate to the requirements imposed by a specific service invitation.

*Usage Context*

The *Usage Context* represents the communication capabilities available to a specific user. It keeps track of the set of terminals the user has registered. The Usage Context has been extended to provide a reference to the current location of the user (cf. User Location) and the associated Local Context.

The Usage Context selects a terminal between the set of registered terminals according to requirements of the inviting service. In case no appropriate terminal registration is available, the location registration information produced with the help of the User Location CO and the LCxt CO will be retrieved. The Usage Context maintains the dynamically changing association to both former components and decides when to access both.

*Terminal Equipment Agent*

The Terminal Equipment Agent contains information about capabilities and current state of a terminal, i.e. a physical device able to handle certain media in a specific way. In its specific form within this project (SAP, Service Access Point), the Terminal Equipment Agent is extended with additional interfaces, enabling the dynamic inquiry of properties of represented terminals.

*Resource Configurator*

The Resource Configurator is the coordinating instance for selection, configuration, and composition of all terminals, media converter functionalities (represented as Mediator Unit SAPs, see below), Service Gateways, etc. assigned to a certain location. In particular, this component is able to configure chains of multiple converters, to evaluate their Quality of Service, and to select the chain most appropriate for the desired task (see Fig. 5). The result of the configuration is a dynamically generated converter chain with stream interfaces for the appropriate media.

*Mediator Unit SAP*

This component represents the functionality of media converters (described below), based on uniform definitions of a Converter Framework. Each converter is subordinated its specific MSAP. It has been designed for the purpose of dynamic binding of converters. The MSAP provides the knowledge about individual properties of its converter, regarding purpose, range of configurability, parametrization, and its QoS parameters. In case the converter itself is not conform to TINA/CORBA, the MSAP can be used as a wrapper to provide such an interface.

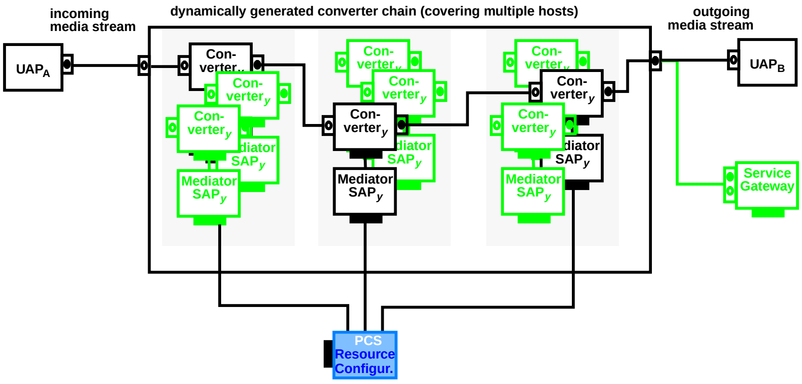



Fig. 12. Converter chain, configured for a specific task

*Medium Type and Medium Format Converters*

The design of all Media Converters and Format Converters is based on the open Converter Framework. The various converters can be distributed in a local environment, or even offered as remote services. This distribution is hidden by the Resource Configurator.

Generally, all converters have two types of interface:

- an operational interface towards the Resource Configurator (through the MSAP),
- one or more stream interfaces for incoming and outgoing information flows.

These interfaces are being defined as a bundled Generic Converter Interface (GCI), providing a uniform and flexible access to any Media Converter or Format Converter. The foreseeing definition of this interface is a major key for achieving flexibility to integrate more converters in future developments. The core conversion libraries are adaptations of products which are available commercially or in the public domain.

Specific examples for converters for first implementation are:

- Audio format converters
- Image format converters
- Video format converters
- Text format converters
- Text-to-Speech
- Optical character recognition
- Speech recognition

The complexity of these conversions is highly different, as discussed in the previous sections.

*Message Store*

This internal service component (cf. Figure 13) provides storage and provision of bulky data of multimedia messages. Such data can be delivered from the *phone.in* service gateway, or from recording tools for customized messages.

*Service Gateways*

Service Gateways are tools for realizing Service Interworking. They are responsible for transporting information into and out of the context of the iPCSS, i.e. connecting the iPCSS to the world outside the TINA platform. They have to consider the specific properties of the connected information and communication services. In particular, they have to adapt the signalling and digitizing of legacy telecommunication systems.

The connections might be synchronous or asynchronous, and it might be unidirectional (simplex) or bidirectional (duplex).

Specific Service Gateways are:

- *phone.in*, *phone.out*, and *phone.duplex* for voice connection with the public telephone network; used for recording voice-box messages, delivering prepared audio data, or synchronously connecting phone talks, respectively,
- *fax.out* and *fax.in* for sending and reception of G3 faxes,
- *mail.in* and *mail.out* for reception and delivery of multimedia e-mail,
- *paging.out* and *sms.out* for delivering signalling information to pagers and short message services,
- *mmc*, for support of multimedia conferencing.

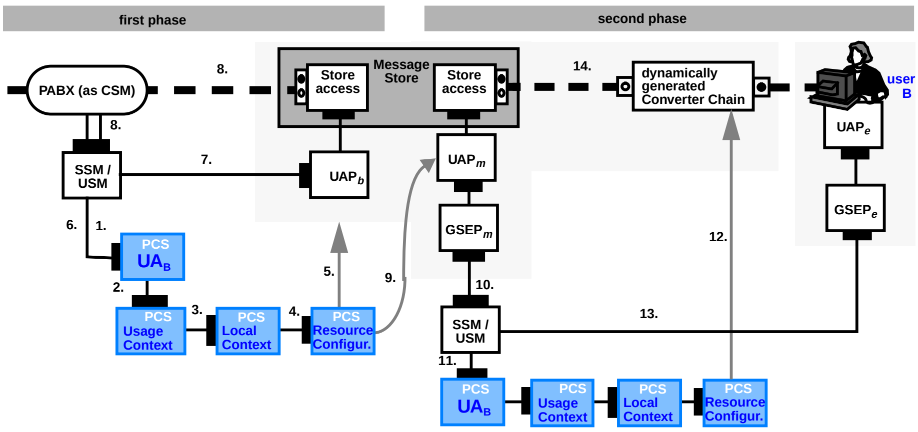



Fig. 13. PCS-enhanced TINA Access Session for an asynchronous communication example

## 6.4. iPCSS Component Interactions

- *Registration*

In contrast to the registration at terminals, the iPCSS enables user to register with locations, e.g. offices, in many different ways, mainly manual registration, scheduled registration and automatic registration. The first two registration types can be realized via a specialized *"User Registration Application"*, representing an UAP CO which interacts with the PCS-UA, which in turn interacts with the User Location CO and/or Personal Schedule CO.

In case of automatic registration, i.e. if the user wears an Active Badge [9], no explicit user action for the registration is required, since an *"Electronic Location System"*, modelled as a specific PCS support application will directly manipulate the user registration data in the User Location CO, i.e. modify the references to a corresponding Local Context.

- *Synchronous communication*

Figure 11 depicts the CO interactions within a PCS-enhanced Access Session. An invitation of user B is delegated via user A's $UAP_x$, $GSEP_x$, $USM_A$ and the SSM to the user B's PCS-UA (1, 2, 3, 4). Within the enhanced Access Session the $PCS\text{-}UA_B$ checks first the Personal Call Logic of user B for the reaction to that invitation (5). In the case that the Personal Call Logic indicates to accept the call, the subscription information of user B for the requested service is checked by the respective SubAgt (6). With service information received from the SubAgt, the configuration available for the user has to be retrieved (7). The current location of user B has to be retrieved by querying the User Location (8).

Using the location information, the LCxt attached to that location can be found (8b). The LCxt component allows terminal equipment suited to the service to be selected (8c). In case there is no terminal available supporting the requested media type, the Resource Configurator (8d) is invoked. Considering the knowledge of the MSAPs about the subordinated converters, it selects and configures one or multiple converters dynamically to an appropriate converter chain (for more details cf. Figure 12). The latter is instantiated as an object with stream interfaces (8e). Its reference is then passed back to the SSM (via LCxt, UCxt, UA). The subordinated Connection Session Manager (8f) uses this reference for the connection of the streams to the converter object.

Considering the personal preferences (9), the communication request is indicated to user B (10, 11). If user B accepts the invitation (12, 13), the SF is instructed to create the corresponding $USM_B$ (14). The subsequent service processing establishes the stream connection, with respect to the possible converter chain.

While the previous description fits most for synchronous communication, e.g. a telephone user participating a multimedia conference, an example for the asynchronous case is given below.

- *Asynchronous communication*

The asynchronous illustration (Figure 13) is explained for the example of an incoming phone call, which stores its message in a voice box due to unreachability of the invited user. As in the examples before, the primary invitation is passed via the PABX and the SSM to user B's UA (1). Her personal logic indicates unreachability, and points to the message store (2, 3, 4). The respective Resource Configurator initilizes the User Application of the Message Store (5), and passes its reference back to the SSM (6). The latter (which is for simplification not separated from the USM) establishes the connection to the UAP (7) and connects the stream to the Message Store interface (8). The store works as voice box and records the message, then the call is terminated.

However, the Resource Configurator (RC) has learned the preference of the invited user to get the voice messages forwarded as multimedia e-mail, either in audio form, or even converted to text by speech recognition. In any of these cases, format or media conversion is required.

An important task of the RC is therefore to *trigger* the User Application in the Message Store responsible for forwarding the messages (9). This UAP begins to establish a new call situation, employing the respective SSM/USM (10), and user B's UA again (11). But now the RC creates an appropriate converter chain for the desired task (12), so that the SSM can establish the connection towards the user's Mail-UAP (13, 14), or, alternatively, a service gateway for forwarding a mail anywhere else.

## 7. Conclusions

This paper has presented an overview of the iPCSS, representing a CORBA/TINA-based platform for the provision of full PCS capabilities as an usage example of automatically configurable technology of media conversion. This technology has been evaluated from the theoretical viewpoint, and quality aspects for the automatic process have been considerd.

As shown in the paper the iPCSS provides enhanced reachability of users while the users are able to control/manage their reachability. PCS capabilities are service generic and are mostly related to the user's access to services.

A prototype implementation of the presented platform will be available by the end of 1996. Further information about the iPCSS can be obtained from "http://www.fokus.gmd.de/ice/".

## 8. Acronyms

| | |
|---|---|
| CO | Computational Object |
| CPE | Customer Premises Equipment |
| CSM | Connection Session Manager |
| GSI | Generic Converter Interface |
| GSEP | Generic Session Endpoint |
| IN | Intelligent Network |
| iPCSS | intelligent Personal Communication Support System |
| LCxt | Local Context |
| MSAP | Mediator Service Access Point |
| OCR | Optical Character Recognition |
| PABX | Private Automatic Branch Exchange |
| PCS | Personal Communication Support |
| POTS | Plain Old Telephone Service |
| PPrf | Personal Profile |
| SSM | Service Session Manager |
| SubAgt | Subscription Agent |
| TE-A | Terminal Equipment Agent |
| TINA | Telecommunic. Information Networking Architecture |
| TMN | Telecommunication Management Network |
| TTS | Text-To-Speech conversion |
| UA | User Agent |
| UAP | (End) User Application |
| UCxt | Usage Context |
| UPT | Universal Personal Telecommunication |
| USM | User (Service) Session Manager |

## 9. References

[1] Schmandt, Chris: Multimedia Nomadic Services on Today's Hardware. - in IEEE Network, Sept./Oct. 1994, pp. 12-21

[2] IEEE Communications Magazine, Special Issue on Computer Telephony, Vol. 34 No. 4, April 1996

[3] Weiser, M: Some Computer Science Issues in Ubiquitous Computing. - in: Communications of the ACM, Vol. 6, No. 7, July 1993

[4] Guntermann, M.; et al.: Integration of Advanced Communication Services in the Personal Services Communication Space - A Realisation Study. - in: Proc. of the RACE International Conference on Intelligence in Broadband Service and Networks (IS&N) 1993 (Mobilise), pp. II/1/p.1-12

[5] ITU-T Draft Recommendation F.851: Universal Personal Telecommunications - Service Principles and Operational Provision. November 1991

[6] ITU-T Recommendations Q.121x series: Intelligent Network Capability Set 1. Geneva, 1995

[7] Dupuy, F; Nilsson, G; Inoue, Y: The TINA Consortium: Towards Networking Telecommunications Information Services. - in: Proc. XV. Intern. Switching Sympos., ISS '95, Berlin, Germany, April 1995

[8] ITU-T Recommend. M.3010: Principles of a Telecommunications Management Network. Geneva, 1992

[9] Harter, A; Hopper, A: A Distributed Location System for the Active Office, - in: IEEE Network, Special Issue on Distributed Applications for Telecommunications, January 1994

[10] Eckardt, T; Magedanz, T; Popescu-Zeletin, R: Application of X.500 and X.700 Standards for Supporting Personal Communication in Distributed Computing Environments. - in: Proc. of the 5th IEEE Computer Society Workshop on Future Trends of Distributed Computing Systems, Cheju Island, Korea, August 1995

[11] Eckardt, T; Magedanz, T; Ulbricht, C; Popescu-Zeletin, R: Generic Personal Communications Support for Open Service Environments. IFIP World Conference on Mobile Communications, Canberra, Australia, September 1996

[12] Magedanz, T: TINA - Architectural Basis for Future Telecommunications Services. - in: Computer Communications Magazine, to appear in late 1996

[13] TINA-C Doc. No. TB_MDC.018_1.0_94: Overall Concepts and Principles of TINA. February 1995

[14] TINA-C Doc. No. TB_MDC.012_2.0_94: Service Architecture. March 1995

[15] TINA-C Doc. No. TB_HK.002_1.0_94: Service Component Specifications. March 1995

[16] Mowbray, Thomas J.; Zahavi, Ron: The essential CORBA, System Integration using Distributed Objects, Jon Wiley & Sons, Inc., 1995, ISBN 0-471-10611-9

[17] Eckardt, T; Magedanz, T; Schulz, M; Stapf, M: Personal Communications Support within the TINA Service Architecture - A new TINA-C Auxiliary Project. - in: Proc. of 6th TINA Conference, Heidelberg, Germany, September, 1996

[18] BERKOM Project "Intelligent Personal Communication Support System", Deliverable 0: Design criteria and preliminary description of functionality. - German National Research Center for Information Technology (GMD), Research Institute for Open Communications System (FOKUS), December 1995

[19] Deutsche Telekom Project "Personal Communications Support in TINA", Report No. 1. - German National Research Center for Information Technology (GMD), Research Institute for Open Communications System (FOKUS), June 1996

[20] Jayant, N.S.; Noll, P.: Digital Coding of Waveforms. - Prentice Hall: Englewood Cliffs, 1984

[21] Doermann, David; Yao, Shee: Generating Synthetic Data for Text Analysis Systems. - for: Proc. of the 4th Symposium on Document Analysis and Information Retrieval, SDAIR, Las Vegas, Apr. 24-26, 1995

[22] Jainschigg, John: Text-to-Speech. Sobering up the "drunken swede". - in: Teleconnect, May 1995, pp. 125-128

[23] Léwy, Nicolas; Hornstein, Thomas: - Text-to-Speech Technology. A Survey of German Speech Synthesis Systems. - UIBLAB Technical report 94.10.2, ftp://ftp.uiblab.ubs.ch/pub/paper/GermTTS.ps.Z

[24] Pfeifer, T.: Speech Synthesis in the Intelligent Personal Communication Support System (IPCSS). - in: Bateman, J. (ed.): Proceedings of the 2nd 'Speak!' Workshop on Speech Generation in Multimodal Information Systems and Practical Applications, 2. - 3. November 1995, Darmstadt, GMD-IPSI

[25] Rabiner, L.; Juang, B.-H.: Fundamentals of Speech Recognition. - Prentice-Hall: Englewood Cl., 1993

[26] Gales, M..J.F.; Young S.J.: Segmental Hidden Markov Models for Speech Recognition. - Proc Eurospeech '93, Berlin, pp. 1579-1582, Sept. 1993

[27] Hoefker; Hoehne; Jesorsky: Automatische Sprechererkennung mit Computern, BMFT Bonn, Forschungsbericht DV 80-002, 1980, cs: BMFT Bonn

[28] International Standard ISO/IEC 11172. Information Technology - Coding of moving pictures and accociated audio for digital storage media up to 1.5 MBit/s (MPEG-1). - Genf: ISO, 1993

[29] International Standard ISO/IEC 13818. Information Technology - Generic coding of moving pictures and accociated audio information (MPEG-2). - Genf: ISO, 1994

[30] CCITT Recommendat. H.261. Video codec for audiovisual services at p x 64 kBit/s. Genf: CCITT, 1990

[31] International Standard ISO/IEC 10918-1. Information Technology - Digital compression and coding of continuous-tone still images. Part 1: Requirements and guidelines (JPEG). - Genf: ISO, 1991

[32] Murray, James: Encyclopaedia of Graphics File Formats. - O'Reilly: New York, 1994

[33] Pfeifer, T.; Gadegast, F.; Magedanz, T.: Applying Quality-of-Service Parametrization for Medium-to-medium Conversion. - Proc. of the 8th IEEE Workshop on Local and Metropolitan Area Networks, Potsdam, Aug 25-28, 1996, publ.: Los Alamitos (USA), IEEE Computer Society Press