A First Look at Delivery of Information Services through Multiple Channels

Norwegian Computing Center
APPLIED RESEARCH AND DEVELOPMENT
Norwegian Computing Center / Applied Research and Development

NOTAT/NOTE

IMEDIA/01/01

Arve Larsen

Oslo
January 2001

Norwegian Computing Center
APPLIED RESEARCH AND DEVELOPMENT

Notat / Note

Tittel/Title:
A First Look At Delivery of Information Services through Multiple Channels

Dato/Date: January
År/Year: 2001
Notat nr/
Note no: IMEDIA/01/01

Forfatter/Authors:
Arve Larsen

- **Sammendrag**/Abstract:

This document reports upon an initial investigation concerning the challenges and requirements in building information services accessible through multiple channels. As a point-of-departure, we selected a desktop PC and a (simulated) WAP-enabled phone as the end-points of two different channels. Furthermore, we have focussed upon a specific type of service: that is, a service related to finding, viewing and completing tax forms. Assuming the need to supply such functionality, this study looks into issues related to creating efficient services, as well as efficient service creation.

Emneord/Keywords: multi-channel information services, WWW, WAP, user interface, service navigation, retrieval, input

Sensitivitet/Sensitivity: Non-sensitive information

Tilgjengelighet/Availability: *Restricted* to channel S Partners and NR until May 2001; *Open* thereafter.

Prosjektnr./Project no.: channel S 11022

Satsningsfelt/Research field: service architecture and service channeling, WWW, WAP

Antall sider/No of pages: 14

Norsk Regnesentral / Norwegian Computing Center
Gaustadalléen 23, Postboks 114 Blindern, 0314 Oslo, Norway
Telefon 22 85 25 00, telefax 22 69 76 60

A First Look at Delivery of Information Services through Multiple Channels

Arve Larsen

Norsk Regnesentral
January 2001

Table of Contents

1. Introduction

The purpose of this document is to present the results of Project B within the *channel S* program [1] in 2000. This project had the title: "Information Services via Multiple Platforms".

One goal of the *channel S* program is to develop a Multimedia Multi-Channel infrastructure, or M3Ci for short. An important part of such an infrastructure is the support for building useful information services.

The goal of Project B in 2000 was to start investigating the challenges and requirements in building information services accessible through multiple channels. As our starting point we selected a desktop PC and a (simulated) WAP-enabled phone as the *end-points* of two different channels. On the PC, the services were assumed to be delivered via web; on the WAP-phone, via WAP.

For our purposes, the web represents a well-known architecture where new and complex information services are being built. In our case, WAP represents the opposite. WAP devices and the underlying mechanisms present a platform for building services which is still undergoing change. Similarly, the WAP-devices themselves are radically different from most well known platforms such as the desktop, the web and even PDAs.

The remainder of this document is structured as follows: Chapter 2 gives a brief introduction to our concept of information services and channels, as well as the problem in focus. In chapter 3 we describe an example of a service with specific implementations utilising web and WAP. Chapter 4 presents our conclusions as well as plans and ideas for future work.

2. Focus of the Work

2.1 Information Services and channels

Our concept of Information Services can be described as interactive access to information and data resources. This includes navigation within the service and its data structures as well as retrieval and input of data.

In order to describe the challenges in delivering information services to multiple devices, we have focused on some basic aspects of such services. These aspects have been selected in order to illustrate the impact which different devices and their associated channels have upon the design and efficiency of the service.

In [5] we have defined the term 'channel' as follows:

> *A channel (or 'service channel') is here said to consist of all hardware and software elements utilised in the provision of and/or interaction with a service. Depending upon the organisation of a service, a service channel can include a number of layers; the primary kinds of channel layers are the device layer, the application layer, the middleware layer (where servers are found) and the network layer. Here, a channel is said not to include the end-user nor the explicit content delivered or exchanged within a service.*

In a nutshell, electronic services are characterised by interaction, where interaction concerns the exchange of information / content through (certain layers of) a service channel. To utilise electronic services, end-users employ devices driven by applications that carry out service logic.

2.2 Problem in Focus

In this work, we have focus on a specific type of service: that is, a service related to finding, viewing and completing tax forms.

For navigation and browsing we focus on the structure of the service and its content. We assume that a main goal of a service developer is to present the structure in such a way that a user of the service will be able to utilise this structure.

For our selected devices (i.e., desktop PC and WAP-phone), the visual feedback on the device is the main source for the users' perception of the service. Although most users of our selected service will have some prior knowledge of the service domain, the complexity of taxation and the resulting complexity and multitude of forms will usually not be fully known and understood prior to using the service.

For retrieval and input of data we have chosen to focus on the efficiency of the service. This affects the available functionality, as well as the basic retrieval and input of data. We have begun investigating and describing the suitability of different channels in a separate study [2]. In this report we concentrate on efficient retrieval and input of data.

3. Case Study

The case study in this work considers one service, available through web on a desktop PC and WAP. The service in question is based on a factual case from the Norwegian Tax Administration (Skatteetaten) [3]. The service should allow citizens to find and complete their relevant taxation forms through our selected channels. The paper forms themselves may be regarded as a separate channel, but this is not covered here.

The traditional paper-based taxation forms have evolved over many years, resulting in forms that are highly specialised, with names and structures reflecting both the current system as well as features inherited through revisions over the years.

Our case study divides the total service into two parts. First we consider finding and retrieving the relevant forms. Since the number of possible forms is large, this may be considered similar to navigating in a large database. We then consider filling in, revising and completing these forms. This involves data retrieval, input and review.

3.1 Web on a Desktop PC

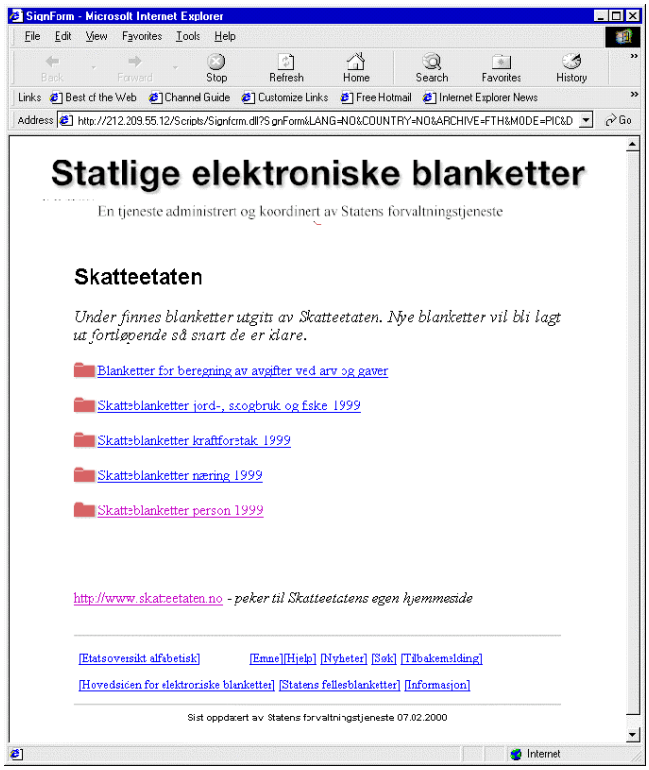
The typical web client represents interaction resources similar to a desktop system. This enables service creator to build on prior knowledge, utilising the resources available both on the client as well as within the service delivery channel. Typically, this involves a large screen and a flexible presentation format. In most cases it also implies a keyboard and a mouse.

Since our service is intended for the general public, we focus on standardised web techniques. The project's original plan specified the development of a prototype which illustrated web as a channel for our selected service. We later decided that it would be more efficient to base such an illustration upon existing services which showed the relevant features and functionality. All of our examples are based on a standard browser, utilising HTML and Java where appropriate.

3.1.1 Navigation

Our first example of navigation within a web-based service is taken from the official web site for Norwegian government forms. This service is co-ordinated by The Government Administration Service (Statens Forvaltningstjeneste), built as a separate service offering public access to forms from several administrations.

The initial access to the service provides one with an initial page where one chooses the administration responsible for the form one wants to retrieve. Further navigation is done be selecting appropriate links on the following web pages.

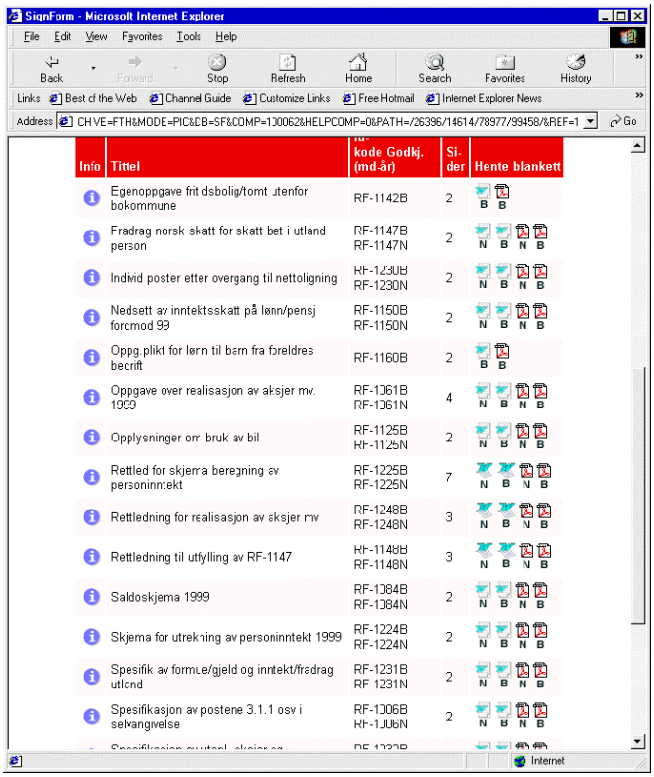


The figure above shows the initial page for taxation, where the total set of forms is divided into five different groups:

1. Declaring inheritance and gifts
2. Taxation of agriculture, fisheries and forestry
3. Taxation of electrical power production
4. Taxation of other companies
5. Taxation of persons

This structure closely reflects the internal structure of the taxation authority, with separate back-end systems within each area. The same correspondence between the organisational structure of an administration or company is often reflected in their web services similar to this.

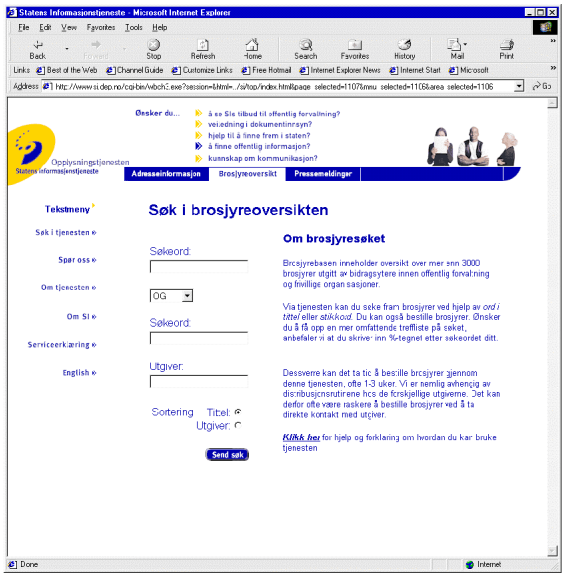
Upon selection of one of these areas, a list of all available forms is presented. An example is shown in the figure below. Most of the forms are available in two formats, Microsoft Word and Adobe Acrobat. In addition most of the forms are available in the two different variants of Norwegian: “bokmål” (B) and “nynorsk” (N).



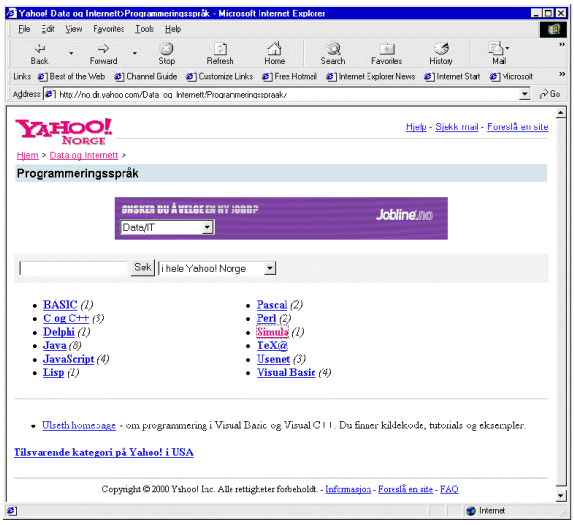
The list shows a link to more information on each form, the official name of the form, its official number, the total amount of pages and links to the different available varieties of the form.

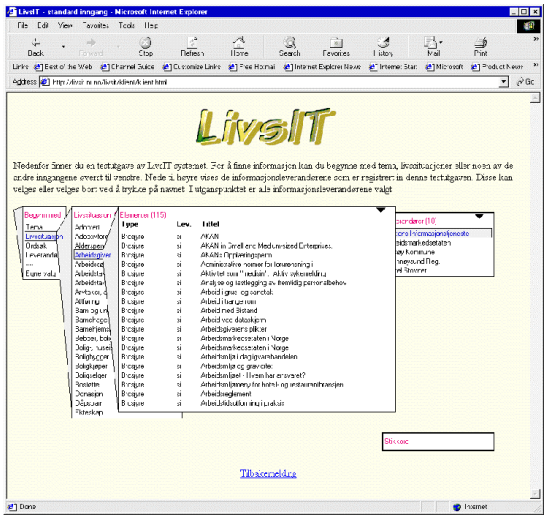
This example is interesting in that it represents one of the most typical ways in which to deliver an existing service through a new channel. Both the existing structures of the old service (the division into five different sections) and the old data (the structure of the forms themselves) are preserved.

The next example utilises the searching facilities available in many electronic services. This service from the Norwegian Central Information Service (Statens Informasjonstjeneste) provides access to a directory of government brochures. The service provides facilities for ordering brochures as well as for viewing brochures that are available online. The screenshot to the left below shows the screen for entering search criteria. The one on the right shows the results of a search.



Another powerful effect often utilised on desktop computers and web-sites is direct visual feedback showing the structure of the service itself. Our first example is a screenshot from Yahoo Norway, showing a page inside the overall structural hierarchy. The page to the left below shows both the current path from the root of the site as well as the possible ways further into the service.



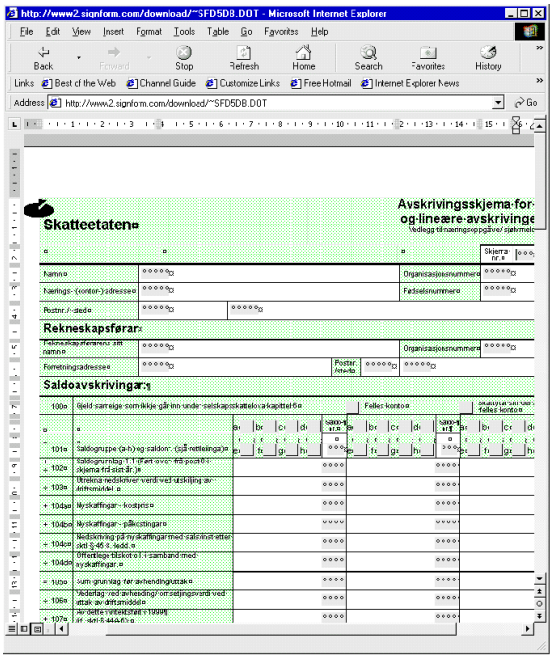


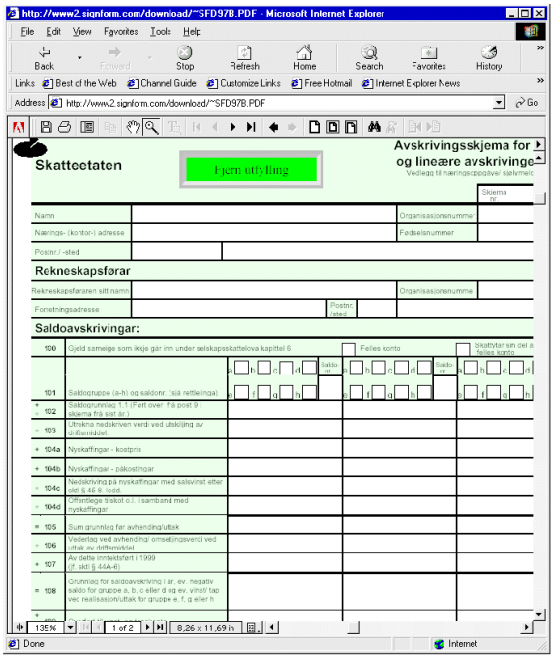
Our last example to the right above is an experimental service which demonstrates access to all governmental information through one portal. The information is structured hierarchically according to pre-defined life-situations. The use of these life-situations presents an emphasis on the users needs and provides users with a structured way of retrieving all relevant information for a given situation. As with Yahoo, this example shows the currently selected path and the ways further. In addition the experimental user interface shows the different possibilities offered along the way.

3.1.2 Data retrieval, review and input

To illustrate data retrieval, review, and input on a web-platform we have chosen three different examples. The first example is taken from the list of tax forms on the net. The example form,

RF-1084, is available in two varieties (B and N) as well as in two formats, Microsoft Word and Adobe Acrobat. This example illustrates electronic forms that are replicas of the original paper form. Both versions enable data input. The technologies used enable forms pre-filled with information from databases. It is also possible to submit these types of forms electronically and to extract the content. These concrete examples, however, are aimed at printing the forms on paper and submitting as a regular paper-form. This type of solution could just as well have been implemented using HTML-forms.





In the SPACE project [4][6], we considered another approach for retrieving, reviewing and inputting data. Instead of focusing on forms as sheets of paper with some information in them, we regarded a form as a collection of data elements grouped in relation to a specific use of the data. The main results were:

- a data-dictionary describing all the possible data elements,
- a language for describing the elements which apply within a specific use context and
- a language for defining user dialogues — dialogues aimed to obtain information about the use context [4][6].

From these elements we created a service for supporting people when moving from one EU country to another. The service allows different administrations to define their data requirements, essentially defining a data structure that acts as a part of a form for the user to fill out. Another part of the service enables administrations to deliver data into this data structure. Our service provided a simple interface for reviewing, changing and complementing the data. The result was a structured data collection where the different data elements could be assembled into subsets corresponding to the required forms.

The design outlined above provides a basis for building flexible forms. A language for defining screen representations could be developed to tailor display according to use context, device capabilities and even user preferences. For devices with limited input and output capabilities, a flexible form where only *relevant* fields are present (and pre-filled wherever possible) could be very useful.

3.2 WAP

The typical WAP client is a mobile phone. Today's mobile phones have their own way of structuring information and their own mode of interaction, making WAP clients distinctly different from web based clients. Efficient information services should be structured in the same way, employing the same mode of interaction. Although WAP clients share the same web-like back-end features, they differ significantly in their operational characteristics. Most important is screen size (pixels/characters), functional buttons and memory size.

In order to illustrate our chosen service delivered through WAP, we built a very shallow prototype illustrating basic navigation and input techniques. This prototype was implemented using flat WML files run through a simulator for a generic WAP-phone. This phone has a four-directional navigation key and three function keys. The figure below shows the generic phone as well as models of two other phones. They illustrate differences in screen size, and the number and types of available functional buttons.

3.2.1 Navigation

In our WAP-example we wanted to show navigation through the same collection of tax-forms as used in some of the examples for web. The first challenge is presenting the hierarchical structure in the first place. The first five-item list provides a nice introduction to the service as well as a hint to its internal structure. However, the available RAM is usually a limiting factor for larger menu structures. Indeed, a simple list showing only the textual names of the forms resulted in a WML-file too big for most current WAP implementations.

Fortunately, since these lists can be ordered, splitting a large list into smaller fragments is easy. For large lists, the ordering can also be used to build a larger hierarchy. For our form service,

we illustrated the same list ordered alphabetically according to its textual name or its official form number.

Conceptually, this fragmentation can be optimised in relation to the size of the list and the amount of available ram on the device. The number of items in a fragment is determined by the amount of available RAM. The total number of items then determines the number of fragments. The final navigation structure is built as a hierarchy of index-files, where the available RAM determines the size of each index file.

Another aspect to consider in this optimisation is the usable size of a fragment. Presenting a long list on a small screen may require copious amounts of scrolling. One some phones scrolling requires one press per line scrolled. For such phones smaller fragments should be considered. Other phones allow for continuous scrolling by keeping a button depressed or by rolling a wheel or something similar. For these phones, larger fragments work quite well.

Another way of approaching this optimisation is to consider the most probable navigation route through the hierarchy in question. Instead of filling the WAP cache with one single page, one could create a stack of pages representing part of the hierarchy. Given that the right part of the hierarchy is represented, this use of a stack of pages could significantly speed up navigation by limiting the number of downloads from the server. On the other hand, if most of the pages in the stack are never used, why include them at all? The policy for deciding exactly which parts of the hierarchy to include in the stack is crucial to this type of optimisation. The possibilities include segmenting the hierarchy manually according to one's best guess, width first, or depth first according to the most frequently used path(s) through the hierarchy. Generally, given some information about the frequency, usefulness or some other quantified measurement of the different nodes, a huge number of well-known tree optimisation algorithms could be utilised.

Given any of these optimisation, the basic navigation through the hierarchy should remain the same.

WAP pages can be structured just as a HTML page, with links embedded in a page of text. Just as with HTML it is customary to present the service structure in some sort of menu. On most WAP devices, this will result in a scrollable list looking somewhat similar to the device's own menus. If the difference is too large, WAP services on that particular device will loose the benefit of a known interaction model.

Navigation through a service consists of navigating through a series of menus. This is relatively straightforward for our type of service, where the main usage scenario is locating a single form or a related group of forms in this hierarchy. A large service would imply a deeper hierarchy. If the user can predict where his target form is located given the presented menu, even large hierarchies should be easy to navigate within. Users would usually discover wrong selections on the first level, limiting the amount of backtracking needed.

The figure below shows some typical navigation screens in our WAP-based service prototype. The first two images illustrate locating the desired form through a list sorted alphabetically by name. The next three images illustrate locating the form through a similar list sorted alphabetically by code.

Navigation by form name ("skjema etter navn"):

Navigation by form code ("skjema etter kode"):

Please note that we have chosen not to include information on the full path selected within the different parts of the navigation hierarchy. While this may be useful in a large-screen scenario, the equivalent information would consume too much of the limited screen area on the WAP platform.

Other services may require different navigation styles. Here, a deep hierarchy may prove unsuitable, especially if the service requires that the user moves between several different parts of the service. Care should be taken in designing the service such that related parts of the services are located within the same (local) region of the navigation structure.

3.2.2 Data retrieval, review and input

The core functionality of our service includes retrieval, review and input of data. In our limited example we have implemented a simple service that could be useful for workers as well as for the Tax Administration. The goal is to fill out a request for a new tax card (skattekort). A Norwegian tax card specifies the main parameters of your personal economy used when calculating your tax payment. This includes expected salary, debts, assets etc. The form for this request is very simple, listing the columns for the user in which the expected values for these parameters should be provided. The request is then signed and returned to the taxation authority, which issues a new tax card. This tax card specifies the monthly tax payments needed in order to pay the correct amount of tax throughout the year.

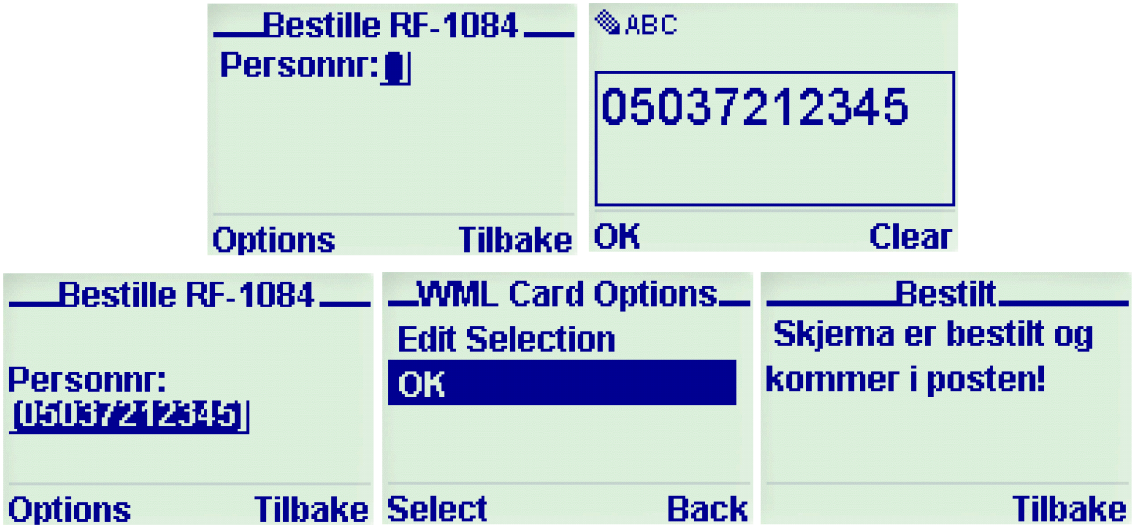
The current paper-based service (*A*) can be described as follows:

1. The user asks for a tax card request form by phone, from the tax administration.
2. The tax administration mails the form (possibly pre-filled) to the user.
3. The user completes the form, signs it and mails it to the tax administration.
4. The tax administration processes the form, issuing a new tax card that is mailed to the user.

In this simple service, the main point is the signature, which must be legally binding. We have defined the following WAP service (*B*):

1. The user locates and selects the tax card request form (as discussed in section 3.2.1).
2. The users provides his or her Person Identification Number (personnummer).
3. The user acknowledges the order for the form.
4. An automatic system at the tax administration mails the form to the user, with the current values for the different parameters presented.
5. The user corrects and/or completes the form with the new values, signs the form and mails it to the taxation administration.

This particular service only automates the ordering of tax card request forms, preserving the other parts of the service. We have focused on minimising the input needed from the user (PIN only). In addition the amount of information presented to the user is kept at a minimum. This simple service was also illustrated for other types of forms. A step-by-step example for ordering (i.e., steps 2-5 above) is shown in the five figures below.



In addition we have defined variations of this service. This variation of the service (*C*) assumes that one can digitally sign the electronic form using one's WAP device:

1. The user locates and selects the tax card request form.
2. The user provides his or her Person Identification Number.
3. The user selects a tax card parameter to change from the list of parameters presented. This list includes the current values for these parameters. This point is repeated for all parameters to be changed.
4. The user acknowledges the changes, which are singed digitally and sent to the administration. The tax card is then automatically processed (possibly with manual checks of randomly selected transactions or transactions with parameter values outside a pre-defined region), resulting in a new tax card being issued.

The service variation outlined below (*D*) is a combination of the two above (*B* and C). It does not require the use of digital signatures, but may still provide a more efficient service than the first two (*A* and *B*) for both the service users and the taxation administration.

1. The user locates and selects the tax card request form.
2. The user provide his or her Person Identification Number.
3. The user selects a tax card parameter to change from the list of parameters presented. This list includes the current values for these parameters. This point is repeated for all parameters to be changed.
4. The user acknowledges the changes. The data is sent to a pre-processing system, which stores the data for later processing, prints a tax card request form with the new information and mails it to the user.
5. The user signs the received form and mails it back to the administration.
6. A tax officer checks the signed form and signals the pre-processing system to issue the new tax card and effectuate the changes in the back end system.

This variety essentially eliminates some key-punching work for the tax officer.

These simple services illustrate the focus on limiting the input needed from the user as well as limiting the information that has to be reviewed. For all these personal forms, the Norwegian Person Identification Number is all that is needed to gain access to the service. Introducing digital signatures or other forms of authorisation and authentication may increase the amount of input needed, depending on the actual security infrastructure.

The amount of data to be reviewed, changed and added varies from service to service. In the SPACE project [4] we investigated similar services with the focus on making the electronic equivalent of paper forms as flexible as possible. By de-composing the forms into their "atomic" data fields and co-ordinating these fields' “universe of discourse” [6], we obtained the basis needed by which to create forms on the fly. This type of solution could be extended in a way that limits the actual fields presented to the user to those fields which are most likely to change. Other fields could then be available through other parts of the service.

In our work here we have looked briefly into the available WAP services. Of the services that reflect services available through web, our work indicates that where the original web service builds on a structured database, the creation of WAP services may require little work. One example is some of the Norwegian newspapers on the web that use databases containing news stories broken into titles, abstracts, short versions and full text versions. However, even though the WAP service may be easy to create, it may not necessarily be efficient for the user.

4. Conclusion and Future Work

Generally we maintain that service channeling should focus upon creating *efficient services* as well as upon *efficient service creation.*

Our brief discussion of delivery of information services through multiple channels indicates that it is possible to provide the same service efficiently through both channels selected in this study (i.e., web and WAP). For web-based services we claim that available technology as well as common ways of building services are sufficient to build services which are efficient for the user. Still, common practices vary so much such that a focus upon creating services which are efficient for users should be maintained.

For WAP services we find that the need to focus on efficient services for the user is even greater. The technology itself imposes limitations on the way services can be built. More importantly, the actual devices vary in such a way that the same service may need to be designed differently for different devices. This suggests that further work is needed in the area of service creation.

Another aspect of service creation which should be explored further is the automatic generation of navigation structures. This type of work could focus on the available RAM and usable size of a single page, as well as look further into the relationship between bundling several pages in relation to the characteristics of the underlying communication infrastructure. For GSM, bundling may prove useful because it may minimise the effects of the long round-trip times. For other communication types such as GPRS, the effects of bundling may be totally different.

In the overall context of this paper, future work should look into efficient creation of services which need minimal adjustment in order to be available through different channels. This should address problems such as the difference between different WAP devices, as well as the fundamental differences between web and WAP. This type of work should also consider other types of end-user technologies such as digital TV. One possible point of focus for this type of work may be to look into the possibilities and restrictions posed by the underlying structure of the service and its data. Highly structured data and services may lead to more flexible services, while less-structured data and services may be easier to develop and maintain.

Our work has focused on strict navigation hierarchies which may be suitable for some services. Other services may require more flexible solutions, for example utilising word searches, associative networks (for instance hyper-text) or similar. More work is needed to see if similar optimisations as bundling, automatic segmentation, etc., may be extended to or integrated within these types of structures.

More work is needed in determining the actual security infrastructures that will be available on the targeted end-user appliances. This is important in relation to the types of services which may be realised, as well as in considering the implications upon the efficiency of the service itself.

5. References

[1] channel S WWW-site: http://www.nr.no/channelS/

[2] Hegna, H. and Holmes, P. D., "Examining the Suitability of Channels for Services and User Roles", NR Technical Note IMEDIA/04/00, December, 2000.
Available for channel S Partners at:
http://www.nr.no/auth/channelS/publications/imedia_2000_04.pdf
Public release (ca. May 2001) at:
http://www.nr.no/channelS/publications/imedia_2000_04.pdf

[3] Norwegian Tax Administration (Skatteetaten) WWW-site:
http://www.skatteetaten.no/

[4] P. Holmes, A. Larsen, S. Myrseth, M. Gritzman: SPACE: An architecture for coordinated intra-European assembly and exchange of citizen data. CE98: Advances in Concurrent Engineering. Proceedings. Tokyo July, 1998. Also available as NR Note IMEDIA/01/98. See http://www.nr.no/publications/imedia_01_98.pdf

[5] Holmes, P. D., Steinbakk, H., and Larsen, A., "The Multimedia Multi-Channel Infrastructure Platform: Scope and Definition", NR Technical Note IMEDIA/05/00.
Available for channel S partners at :
http://www.nr.no/auth/channelS/publications/imedia_2000_05.pdf
Public release (ca. May 2001) at:
http://www.nr.no/channelS/publications/imedia_2000_05.pdf

[6] Larsen, A. and Holmes, P. D.: *An Architecture for Unified Dialogue in Distributed Object Systems*. In TOOLS 26, pages 244-258. IEEE, 1998. Also available as NR Technical Note IMEDIA/03/98. See:
http://www.nr.no/publications/imedia_03_98.pdf