Pers Ubiquit Comput (2004) 8: 1–18
DOI 10.1007/s00779-003-0255-6

ORIGINAL ARTICLE

Dan Chalmers · Naranker Dulay · Morris Sloman

A framework for contextual mediation in mobile and ubiquitous computing applied to the context-aware adaptation of maps

Received: 20 March 2003 / Accepted: 30 September 2003 / Published online: 15 January 2004

Abstract Context has many aspects, which may vary widely, such as the device, environment and user. The perception of data in different contexts also varies widely. We present a new, flexible approach to meeting needs and limits arising from context: *contextual mediation*. In this paper limits are defined as goals over managed system resources. These can be met by the selection of data, taking into account preferences over its semantic and syntactic properties. The specification of this selection is presented in detail and the supporting framework is described. We illustrate our description with examples from a context-aware map application and present experimental results and experiences which demonstrate that contextual mediation enhances the usability of the application in restrictive contexts of use.

Keywords Context awareness · Contextual mediation · Map adaptation · Specification of adaptation

1 Introduction

In ubiquitous computing [1] a wide variety of devices will be in regular use by many people. Many of these devices will suffer limitations which will impact the users' perception of the data presented: their form factor often limits screen size in favour of portability, user input devices lose fine control in favour of robustness and battery life is improved by lowering CPU power, reducing memory size and limiting network connectivity. For laptops and kiosks the limitations may not be as severe as for mobile phones and PDAs in all respects but will still have an impact on use. Current solutions to device limitations tend towards support for a small range of specific devices.

Context extends beyond the device capabilities. The tasks to which devices are applied, e.g., tourism or making deliveries, affects levels of interest in types of information such as museums and house numbers. Activities engaged in while using the device, e.g., driving or talking, affects the amount of information which can be interpreted and the user's interest in types of information such as traffic conditions. Environmental conditions, e.g. background noise or lighting, affects how easily certain media types can be understood.

A general solution to these problems, taking in the wider definition of context, seems desirable—as we discussed in [2]. Managing a wide range of contexts requires a sophisticated approach able to differentiate beyond the most restrictive situations and ideal conditions. Much of the data offered in response to a request is likely to be of little value and there will often be more data available than the user could interpret or the device being used could display, e.g., a request for a map of a given area may return data about many features. Some of this data may be generic, such as roads; some will be of limited appeal, e.g., traffic levels, shop prices and house numbers; and some will be generated dynamically to meet specific needs, e.g. transcoding images for use on a specific screen or translating text language.

Contextual mediation operates by selecting the most appropriate subset of the offered data in order to satisfy a request. This is a form of application-aware adaptation [3] and the principle was mentioned in early work on context awareness [1,4] but has received little detailed treatment in the literature. Different users, in different contexts, will have differing preferences. Mediation must be able to differentiate between useful and unwanted data (with many degrees in between) and use this differentiation to meet goals, such as timely delivery, screen space availability and price limits. Contextual mediation is the process of making this selection, which is distinct from any transcoding being performed on the data.

The term mediation is also used in database and information retrieval research to describe a component which performs integration of data from multiple sources

D. Chalmers (✉) · N. Dulay · M. Sloman
Department of Computing, Imperial College London,
South Kensington Campus, London SW7 2AZ, UK
E-mail: d.chalmers@imperial.ac.uk

and presentation to the user. This function may also be part of the contextual mediation framework, as selecting amongst alternative representations of data from different sources may be considered. We do not address issues of searching, format conversion or ontological translation here.

The range of devices and the contexts in which they are applied will constantly be expanding; therefore, it is undesirable for content providers to support multiple versions of their content, or application designers to produce many versions of a program, each adjusted to a small range of devices or users. A mechanism for general specification of requirements for mediation is needed—this is the core issue addressed in this paper. In particular, these specifications must be able to encode a sophisticated response to variation in many *aspects* of the context, including screen, network speed, a user's task, etc.

Our approach is to provide a generic framework for describing how to select the most appropriate data for a context, which maintains a separation between the application and mediation. We use context aspects to select the most appropriate *profiles* which specify the required mediation rather than trying to cater to all possible context variations. This also has the benefit that conflicting profiles will be resolved in a consistent manner.

To summarise, we are interested in an approach to the selection of data which improves the usability of data in a wide range of contexts. Our approach is general and flexible, supporting improvements in the range of contexts responded to and in the response which the user requires. At the same time, the user is to be shielded from the complexity of specifying this behaviour, which we abstract as context-enabled profiles. The selection forms part of a framework, which includes data description and resource management.

1.1 Requirements for a framework for contextual mediation

The requirements for our framework for contextual mediation are as follows:

- Flexible context specific profiles capable of describing a wide range of user and system level needs and limitations. Profile composition to enable independent profiles, activated by different aspects of context, to describe needs over the same data. Profile specifications that are declarative and can be directly executed.
- A *metadata model* which captures the structure, attributes and semantics of the data. An object-oriented model of semantic types to support this is desirable. Profile specifications should be orthogonal to the data model.
- Provision of *resource management* in order to work within dynamic system limits or shared resources. Resource management will enable limitations to be reported and provide admission control functions to enable goals to be met.
- Application independent mechanisms for profile specification, metadata and resource management are required to allow a general framework to be applied. The mechanism for applying mediation should be application-aware in order to operate within the application's interaction semantics.

2 Case study

One class of applications which are clearly applicable to ubiquitous computing and contextual mediation is the display of maps. We use this as a case study to motivate our examples. Earlier work by us in this area is described in [5].

Map data can be extended with many media types beyond the traditional 2D representation of topology, c.f. [6]. A wide range of media and classes of features of varying utility can be offered to describe an area. Our techniques are generally applicable to many other applications, e.g., Web browsing and news services. The standard vector map formats available provide data with a rich structure typical of many emerging media standards, e.g., MPEG-4 and the many XML encodings, making the techniques described applicable over other media.

Consider a user with the following context: a network of 3.6kB/s and 100ms latency, a screen of 400 × 400 pixels; and a task of navigating in a car on business. A journey into central London is being undertaken. A location service will predict the area the user is likely to need to know about for the next few minutes and will notify the map application. The application will display a map of that area. As the journey progresses towards its destination the context of use changes: the speed decreases as the driver leaves major roads, and the area covered by each map decreases as the navigation becomes more detailed and immediate rather than involving long distance planning.

In Fig. 1 we present a sequence of maps which would be presented by the application towards the end of this journey. On the left we show mediated maps; on the right the same map areas are shown with no mediation. It can be seen that the unmediated map is rather cluttered in the early stages of use, and that the inclusion of unnecessary features obscures some detail and adds to the information to be interpreted (even if it is then ignored). We also see from the notes of time taken (including the metadata loading for the mediated maps) that the mediated maps take less time and a more consistent time; the first two unmediated maps take longer than the given deadline to load (the first map almost double that time).

These maps illustrate the zooming in (row 1 to 2) required in order to allow small detail to be shown clearly. This is a similar principle to moving from a road atlas to a town centre street map. At some point, with any rendering technique or data format, changing the

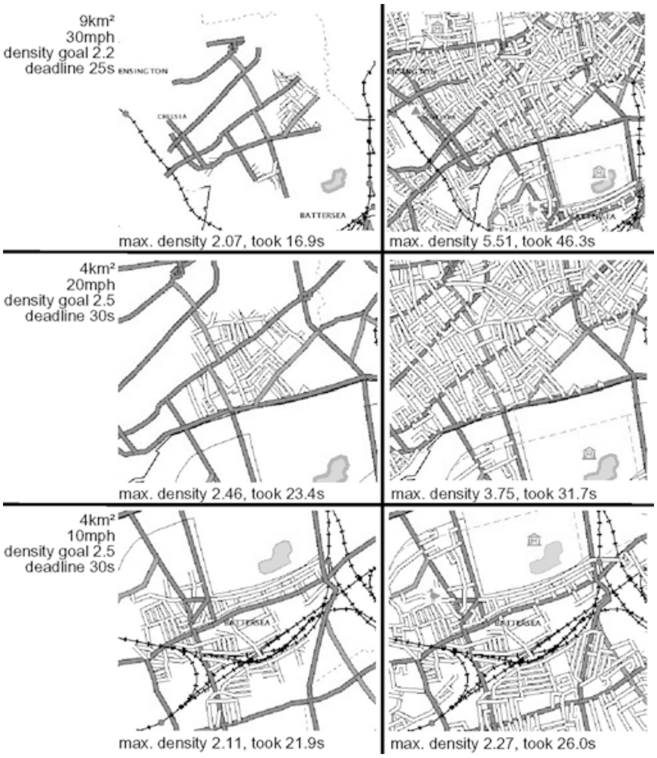

9km²
30mph
density goal 2.2
deadline 25s
KENSINGTON
CHELSEA
BATTERSEA
max. density 2.07, took 16.9s
max. density 5.51, took 46.3s
4km²
20mph
density goal 2.5
deadline 30s
max. density 2.46, took 23.4s
max. density 3.75, took 31.7s
4km²
10mph
density goal 2.5
deadline 30s
max. density 2.11, took 21.9s
max. density 2.27, took 26.0s

Fig. 1 An example journey. Left-hand mediated, right-hand unmediated. All have a 3.6kB/s network, and a 400 × 400 pixel screen

level of detail which is effective requires a change in the scale of the map. Changes in scale may be a deliberate choice for the user as they have such a substantial effect on the interpretation of the map, or come from a location service.

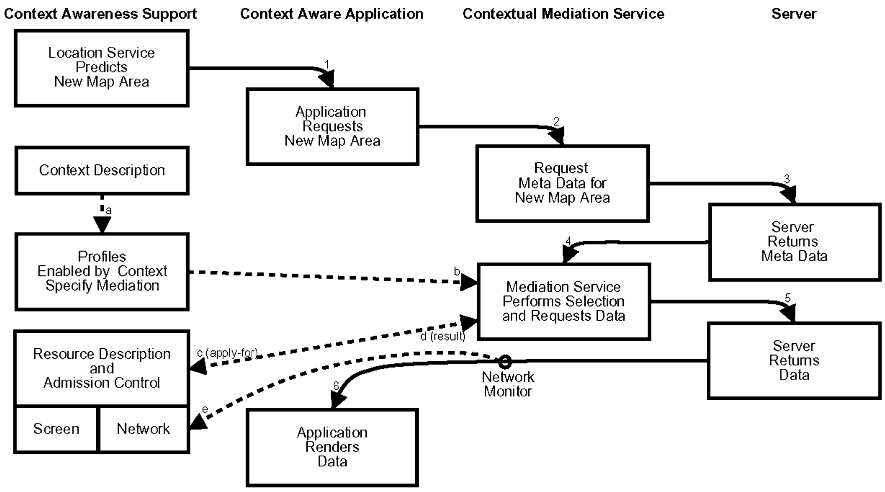
Road names are included in the data we download but are not rendered here. In maps at this scale road names would have to be drawn so small as to be hard to read and would add to clutter on more detailed maps. This clutter and hard-to-read text can often be seen on printed city centre street maps. The maps our application produces are still useful for navigation as much of this process is in the interpretation of symbols and the graph of roads. Specific road names can be made available when appropriate. For instance, on selection of an area a small set of road names may be drawn, e.g., describing roads at a junction. In this case, font size and layout limitations are rather less strenuous and readability is substantially improved. This is preferable to zooming in for cases where a wider view of the road network and other features aids route planning. This approach is similar to that in the paper world where large scale maps generally do not name small features, while town plans name each road but their scale forces the focus onto a smaller location.

Modern in-car navigation systems provide clear directions for a driver already. Our research is addressing a different need: the flexible provision of maps for many purposes. While in-car systems provide usable directions, they do not support a more open-ended "browsing" style of map use, for instance, allowing ad-hoc detours to local attractions. When arriving at a destination a system which supports dynamic data may start to show parking places based on cost, which are of no interest for most of the journey. If a passenger was navigating using a PDA then the map can be carried outside the car and start to provide data relevant to a pedestrian. While we may aspire to the drawing quality of commercial map systems we are interested in *which* data are shown rather than the rendering itself.

3 An overview of the mediation process

The flow of the application's mediated request for data is illustrated in Fig. 2. A location service causes a new map area to be requested by the application. First, metadata (see Sect. 4) describing the available data for that area are loaded. The current context description enables profiles which are used to rate the data offered (see Sect. 5). Resource management holds descriptions of the network and screen. The rating allows the mediation system to make the best selection of a subset of the

Fig. 2 The flow of operations in the mediated map application. Solid lines (1 to 6) show the application flow. Broken lines (a to e) show the interactions supporting contextual mediation


Context Awareness Support
Context Aware Application
Contextual Mediation Service
Server
Location Service Predicts New Map Area
1
Application Requests New Map Area
2
Request Meta Data for New Map Area
3
Server Returns Meta Data
4
Context Description
a
Profiles Enabled by Context Specify Mediation
b
Mediation Service Performs Selection and Requests Data
5
Server Returns Data
d (result)
c (apply-for)
Resource Description and Admission Control
Screen
Network
e
Network Monitor
6
Application Renders Data

data, taking resource considerations into account. The selected map data is then requested and displayed. The map data loading is monitored and used to update the network resource model.

3.1 An overview of profile specifications

Profiles allow the description of needs and limitations arising from context. We shall introduce the form of the specification of a profile although we shall address the details later on. Firstly, we define a specification and give it a name, for convenience here "car_moderate_speed". This profile specifies needs arising from travel at a certain speed.

```
SPEC:car_moderate_speed
M: cMatch(speed) = (21 <= value <= 45)
G: goal(deadline) = 25
U: utilityFn(road) = 1
   utilityFn(road:minor) = 0.9
   utilityFn(road:minor.distance) = prefNearer(value)
   utilityFn(road, map.scale) = mediumDetail(value)
```

The profile starts with a context matching function (M). The profile is enabled when the context aspect "speed" is sensed as being between 21 and 45 mph. In all functions the keyword "value" on the right-hand side refers to the value of the parameter on the left-hand side. We then describe limits arising from the context (G). Speed of travel limits the time available to download map data. This may be due to the location prediction function. Here we specify that the map should be displayed within 25 seconds of receiving the location prediction. In order to meet this deadline it may be that some map data has to be degraded or omitted. We use utility functions (U) to describe preferences amongst the data, firstly, over elements of the map with different semantic content. We specify here that roads are the most important elements in the map, but that minor roads are less important. The third function takes distance as a parameter and indicates that minor roads close to the point of interest are more important than those further away. Finally, we use a utility function to describe preferences for different representations of roads, according to their scale. It is likely that a certain level of detail is best for navigation in this context. More detail may well be lost or distracting at speed. Less detail may not describe small turns in the road which the navigator would find useful.

The differentiation according to both semantics and data attributes allows a degradation of the data to be made in a manner which is sensitive to the context. In this way the goals are met with minimum impact on the user. Note that we describe the user's perception of data rather than the technical process leading to the data. Transcoding proxies, etc., are assumed to exist in the ubiquitous infrastructure and offer data which can fulfil the user's needs. The impact on a user is our metric of success, rather than maximising resource use. In Sect. 5 we shall expand on these definitions and provide further examples.

4 Describing data to mediate

A wide range of data may be offered in response to a request for a map of an area. This may include different features, survey scales, prices, encodings, etc. In order to support the mediation of data, a description of the data which are offered is required. We describe our use of metadata more fully in [7] but summarise the concepts below.

The response to a request we call a *document*. A document may be an area of map, a Web page, etc. The semantic structure of a document is described by *elements*. Elements are instances of a *semantic type*. In a map a feature would be described as an element, e.g., the river Thames, the M1, which would take types of river and road. Our type system allows elements to take multiple types, e.g., a building which is both shops and housing. The types are structured through specialisation, e.g., a "motorway" is a subtype of "road".

The data realising the document we refer to as *variants*. A variant may provide a realisation of one or more elements. Each separate data item which encodes a representation of some element(s) is called a variant, e.g., describing a given road at different scales or in different data formats.

In addition to typing, further description is required in metadata to support mediation. Both elements and variants may have *attributes* describing their properties. Map elements may be described by their location. Variants have attributes which describe the data, such as size in bytes and scale.

We expect data to be structured. Elements may be structured to reflect semantic dependency between elements, such as building names requiring a definition of the building to be meaningful. Elements may be grouped in order to collect similar semantic data together, reducing the overhead of metadata. Variants are structured to reflect dependencies in data, such as common header data used by many variants.

In Fig. 3 we illustrate in UML the relationship between documents, elements and variants, while Fig. 4 illustrates two different representations of a section of the river Thames. All elements have a variant corresponding to omission which may be selected, shown as a third variant here.

Our metadata model is based on [8]. Contextual mediation may be applied independently of metadata encodings, although the separation of elements and variants and the use of semantic typing are important in our work. We use a simple "general type:specific type" syntax for clarity with a common encoding of types between the metadata source and mediator. Work on ontologies and the semantic web [9] may replace our type system in the future.

We use a GZipped NTF vector map data in our prototype, but our approach does not depend on this in any way, although vector map data is more amenable to mediation than raster data. Modifying encoding and

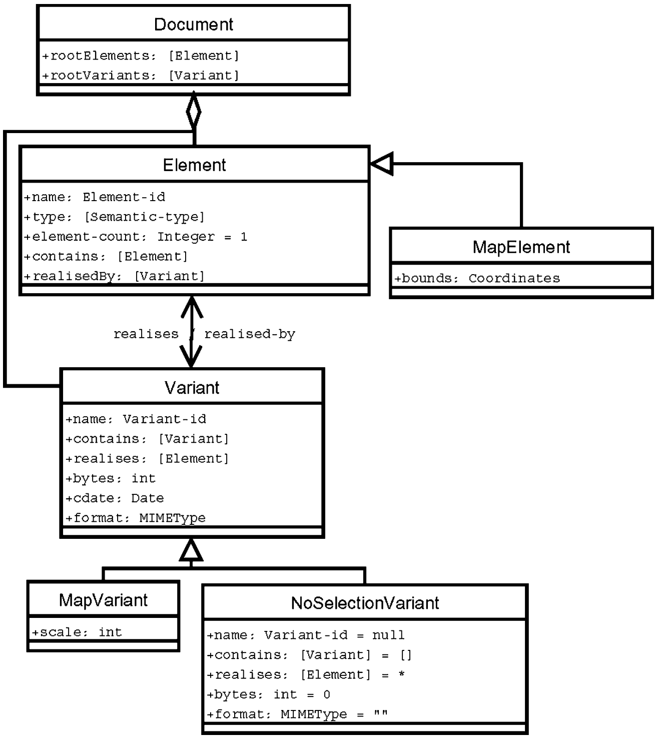


Fig. 3 Metadata for maps

compression alters the data volume, CPU and memory requirements for processing and rendering the data. These changes do not alter the need for mediation techniques.

Metadata may include descriptions of tailored data produced by device specific proxies and arbitrary transcoding produced by general purpose proxies. The intention is that while specific support for devices or tasks may not always exist, it may be used when available.

5 Specifying mediation

In order to make effective selections to support contextual mediation, we need to be able to measure perceived quality. We have to first define which elements are useful and which variants of these elements would be the best for a specific context, while omitting unwanted elements. In the case where the combined resource requirements of the best variants exceed our resource limits (derived from the goals), we have to make the best selection based on the following criteria: which elements are most important, which variants are acceptable and how much degradation the alternative variants cause and whether using them would help meet the resource limits.

An arbitrary decision is unlikely to be the most satisfactory and simply addressing the resource constraint cannot be expected to provide a coherent or satisfactory selection from the user's perspective. We now describe the profile specifications, which enable contextual mediation by allowing this differentiation between alternative elements and variants to be made.

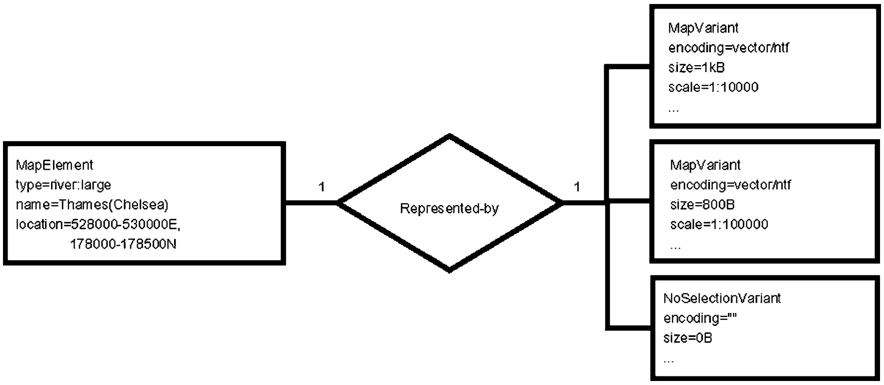
5.1 Utility functions to select amongst elements

The response of users to different semantic types are encoded by associating a *utility* with the type, e.g., to describe a preference for displaying information about roads rather than rivers. The utility is normalised between zero (no interest) and one (the most interesting/ important type of data). The following value assignment define the default utility for any type ("*") to be 0.4, for roads to be 0.85 and for motorways to be 0.8. Utilities for the most specific subtype of the element type are applied, e.g., a utility for "motorway" would be applied rather than that for "road" where the element to be rated is a motorway.

```
utilityFn(*) = 0.4
utilityFn(road) = 0.85
utilityFn(motorway) = 0.8
```

The description above illustrates the use of a static value assigned to elements of a given type. In many cases this is sufficient and clear. However, for some cases the preferences encoded may be adjusted due to attributes of that element. For instance, a railway station closer to the centre of interest may be deemed more interesting than one farther away. In this case it is useful to be able to modify the utility of the elements to reflect this. This is illustrated in Fig. 5, where preferences for buildings, major and minor roads at different distances are illustrated. Here we show minor roads as rapidly becoming

Fig. 4 An example element and variants



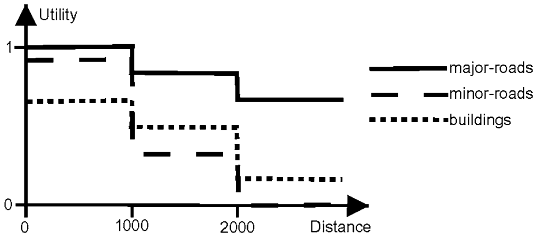


Fig. 5 Example utility functions for distance

less important with distance, while an indication of buildings remains more important. Major roads remain the most important features at all distances, possibly reflecting preferences arising from navigating at high speed or at some distance from the final destination.

The effect of an element's attribute on the perception of different types of element can be reflected by the use of different functions. Different attributes and semantic types may give rise to different utility functions, the first of which we expand in-line to illustrate a discrete function:

```
utilityFn:  Type → utilityFn → 0..1
utilityFn:  Type, EAttribute → utilityFn EAttribute.Type → 0..1
utilityFn(road.distance) = prefNearer(value)
  where prefNearer(dist) =
  |          dist < 1000 = 1
  | 1000 <= dist < 2000 = 0.95
  | 2000 <= dist        = 0.9
utilityFn(house_nos.distance) = strongPrefNearer(value)
utilityFn(road) = 0.85
```

Note that throughout, "value" on the right-hand side of a function refers to the value of the parameter on the left-hand side. For the first two functions above this is the distance of the element from the point of interest. The application of different functions to an element is controlled by the "type.attribute" selection on the left-hand side so different functions are selected for the attribute "distance" for different types of element.

5.2 Utility functions to select amongst variants

Preferences for different variant representations of an element are defined using *utility functions* applied to the attributes of the variants, similar to that in [10, 11]. Again, these are normalised functions, zero indicating the variant is of no use, one indicating that no further technical improvement would be perceived.

Utility functions can be defined relating to scale as depicted in Fig. 6. For cars a lot of detail makes the map harder to read, while a large scale is often still useful. For hikers, there might be less tolerance for the loss of detail of bigger scales, but more time to read and use the greater detail of the smaller scales. Unlike [12] we make no assumption that any offered variant is ideal when rating their utility. In many media there comes a point at which no improvement or an adverse effect is perceived for an increase in the encoded detail. The quality perceived will be due in part to the context of the interaction and cannot generally be extracted directly from the quality of the encoding or data size, as in [13, 14].

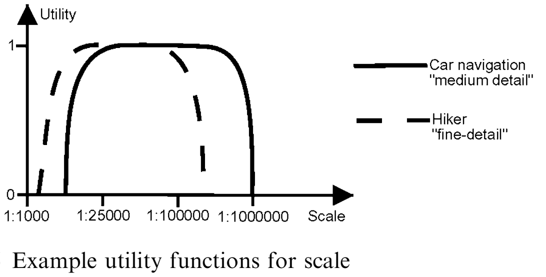


Fig. 6 Example utility functions for scale

The utility function may assign specific values to specific attribute values, e.g., to the language of the tourist information associated with a map. This use of utility has a similar effect to the explicit weights found in some other approaches, although our mechanism is more general.

The attributes over which utility functions are defined will depend in part on the media format which must be at least partly known, e.g., "survey scale" is only a meaningful attribute for map data. Universal attributes include "MIME type" and "creation date". MIME type can be used to express the following example of preferences: when driving a car, audio data may be preferred over graphical data.

The *types* of data a variant represents may be used to select amongst different utility functions to be applied to each attribute, which reflect the different ways in which the perceived utility of the data varies. Utility functions are selected with a function of the form:

```
utilityFn:  Type, Format → utilityFn:  Format.Type → 0..1
utilityFn:  Type, Format, VAttribute →
            utilityFn:  VAttribute.Type → 0..1
```

Utility functions applied to variants take two parameters: the elements' semantic type and the syntactic attribute of the variant the function is defined over. Different attributes, formats and element types may give rise to different utility functions. The utility functions over scale, shown below, describe a perception that the loss of utility with increasing scale is greater for house numbers and traffic than for roads. The utility function over the survey date would indicate that up-to-date data are to be preferred for all element types. Finally, we say that for all element types and formats we prefer vector data over raster. The value passed into prefVectorOverRaster is the value of the format (map/image or map/raster). Again, "value" on the right hand side is the value of the attribute on the left (not the type).

```
utilityFn(road, map.scale) = mediumDetail(value)
utilityFn(house_nos, map.scale) = fineDetail(value)
utilityFn(traffic, map.scale) = fineDetail(value)
utilityFn(*, map.survey_date) = prefRecent(value)
utilityFn(*, *) = prefVectorOverRaster(value)
```

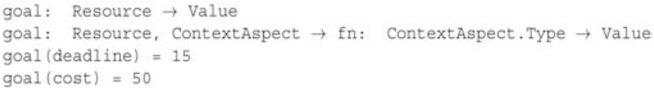
5.3 Resource management

A fundamental need in contextual mediation is to work within limited resources. The first resource we considered was time. A key factor in the speed of response to a request is the network and server latency and throughput. Given an estimate of expected network and data characteristics the total time required to serve a set of requests can be estimated. The mediation process may then be able to limit the data requested in order to (attempt to) meet a deadline. Our network model, which is not described in detail, maintains a model of the round trip time and throughput by instrumenting data loads (rather than creating measurement traffic) [15]. It is tuned to provide stability in the face of minor changes and also an agile response to major changes in performance and has similarities to that in [16, 17] but is implemented at the application level. A time-sliced bandwidth reservation system provides admission control.

We have also been working on a model of how data display consumes screen space. For a given screen resolution, different data may be appropriate in order to avoid the need to scroll to view the complete rendering. Where a screen is physically smaller or has low resolution a given rendering may become harder to read. Maps may also include hyperlinks but the accuracy of selection needs to be considered—it may be desirable to limit the number of links under a finger-tip to an easy choice. Where the ambient light falling on the screen is bright or the screen is vibrating the display may become harder to read. By controlling the density of data displayed on the screen readability can be improved. We have a model of the estimated use of pixels in areas of the screen. The estimated pixel use comes from metadata describing map features, and the application describes the size of the drawing area available and the rendering to be used. We divide the screen into sub-areas in order to consider localised feature density on the screen. We refer to "screen density" in this paper, which is a metric for the expected use of a sub-area of the screen in rendering data. A higher density indicates more information being presented in a given screen area.

5.4 Goal functions

We have goals, or constraints, on modelled resources to be applied over the whole document retrieval, but we make no guarantees that they will be met. We have used goal functions for download deadline, maximum data density on screen and maximum cost in the map application to date. A goal may be represented in a similar way to utility functions:



```
goal:  Resource → Value
goal:  Resource, ContextAspect → fn:  ContextAspect.Type → Value
goal(deadline) = 15
goal(cost) = 50
```

It is also possible to define goals as functions in terms of other values known during the selection process (which we treat as context aspects here). For example, we know the area of the map requested. If the goal is a function of the map area then a selection containing data which can be cached and reused later (e.g., after zooming in) may be given longer to download. Similarly, a high speed may limit the download time (by reducing any fixed deadline).

```
goal(deadline, map_area) = 0.2*value // in km^2
goal(deadline, speed) = -value/5     // in km/h
```

5.5 Executing specifications

We will not discuss selection mechanisms in detail in this paper. However, we assume that the overall flow of interactions is similar to that in Fig. 2. Initially, all elements with a non-zero utility will have their highest utility variant selected. Where goals are defined but there is a resource shortfall the selection process follows an algorithm which treats differing elements according to their utilities, and selects amongst the variants of each element according to their utilities. The algorithm maximises the perceived utility to the user, within the resources available given the goals. We provide an overview of the application of a profile to form a selection in Fig. 7. Here we illustrate the profile being applied to the metadata and the selections being made. The *degradation path* will reflect the utilities of the data, such that the least important data will be offered for degradation first. This is by selecting the lower utility variants of elements, including a zero utility omission variant. Each element is represented by exactly one selected variant in any data request.

An alternative to a fixed selection would be to use the degradation path to provide an order for a sequence of requests. However, we did not follow this approach because the overheads (from processing, request messages and header data) increase for each request dispatched; also a model of the network would still be required in order to calculate the last request which may be made (especially where long round trip times are experienced). Additionally, if the map is drawn incrementally (so it shows the most important elements first) it may not be clear to the user which drawings are incremental and which show the final data, as the map would tend to build in patches (due to grouped features and an association with regional header data) as well as in layers of differing utility features.

Our selection mechanism is located on the client side, along with resource models, context state models, profile management and vector map rendering. This simplifies some issues such as proxy discovery, privacy, etc., for our prototype. In the general case the selection (or other functions) may be performed on proxies in the network where this is appropriate, e.g., where the client has a slow last-hop network, or is limited in processing power or memory.

5.6 Profiles for context

As discussed in [14, 18] there is a need to deliver data at a quality which enables the user to perform his task in the given context. Aspects of context include: location, speed of movement, task, activity and screen size [19].

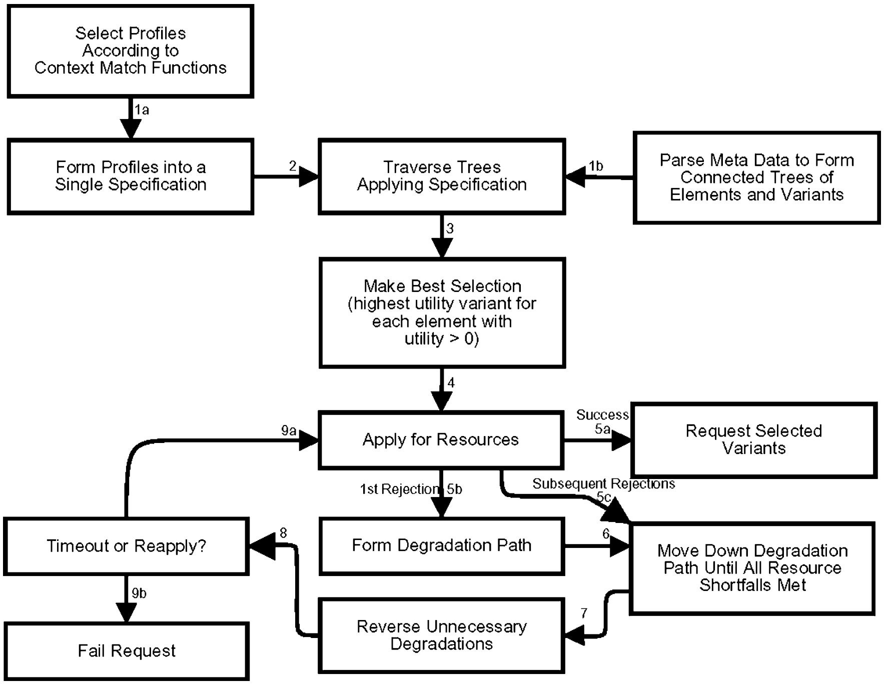

Select Profiles According to Context Match Functions
1a
Form Profiles into a Single Specification
2
Traverse Trees Applying Specification
1b
Parse Meta Data to Form Connected Trees of Elements and Variants
3
Make Best Selection (highest utility variant for each element with utility > 0)
4
9a
Apply for Resources
Success
5a
Request Selected Variants
1st Rejection 5b
Subsequent Rejections
5c
Timeout or Reapply?
8
Form Degradation Path
6
Move Down Degradation Path Until All Resource Shortfalls Met
9b
7
Reverse Unnecessary Degradations
Fail Request

Fig. 7 Applying profiles and performing mediation

Each of these *context aspects* will have some effect on the overall profile. A user may well encounter various combinations of context aspects, e.g., using several devices and undertaking various tasks with each. Clearly, anticipating all combinations of context aspects which might be met and defining profiles tailored for each is not practical. Instead, we define a comparatively small number of profiles for commonly used devices (e.g., desktop, laptop, PDA, mobile phone) specific tasks (e.g., tourism, job functions, shopping, commuting) and for co-location with groups of people (e.g., colleagues, customers, in-public, alone). Profiles may also be written for contexts which are defined by combining context information from different aspects, e.g., driving a car slowly, or fast.

A profile will be applicable when one or more context aspects have certain values, or are within or beyond some value. We use *matching functions* to achieve this. The specification for the profile may contain a matching function for each of a subset of those aspects in the total (known) prevailing context:

```
cMatch: ContextAspect → matchFn: ContextAspect.Type → Boolean
cMatch(task) = (value == delivery_driver)
```

A profile is the combination of its context conditions and the specification which describes the response to that context (utility functions and goals). Any function class (utility over element, utility over variant or goal) may be omitted from a profile where there is no applicable function as a response to the context aspect(s) matched. Context matches may be omitted for basic profiles which are always valid. So:

```
Spec: [cMatch*] → ([utilityFn*], [goal*])
```

Whether the profile is applicable depends on the context conditions being met. This may not be as simple as an equality relation as the aspect may not take a simple numeric type—it may describe abstract notions, e.g., in-car, on business, etc., or geographic spaces, using coordinates. Profiles are applied if *all* the context conditions are met according to their matching functions. A default matching function result may be defined, to support operations despite incomplete context data (the default is false unless defined).

Multiple profiles will be enabled at any one time and so will give rise to conflicting specifications. We form profiles into OO-style inheritance hierarchies (at run-time) in order to resolve these conflicts. Inheritance is derived from context-matching functions. If one profile's matching functions define a subset of another's, by matching on more aspects or with more specific matches, then that profile takes precedence. Where an aspect takes a range of numerical values the value range must be equal to or within the general one, e.g., $20 < speed < 30$ is more specific than $10 < speed < 40$. Where the context takes a value from a tree of conceptual values the value in the specialised profile must be equal to or a descendant of the one in the general profile, e.g., location = my_office is more specific than location = office_building. Specialisation may be achieved by matching on an aspect which is not used in the general profile, e.g., "fast and in a car" is more specific than "in a car". The specialised profile must be more specific than the general one in *at least one aspect*. Functions in the more general profile are retained where there is no conflict, as for inheritance in OO programming.

In the map application we apply mediation to meet download deadlines and screen density limits. The use of utilities allows us to control feature inclusion, which is the main effect used in meeting goals. We do see some variation in variants but the effect is subtle in most cases. To illustrate the reflection of contextual needs and limitations, consider the following profiles, which are extracts from those in use in our prototype application, presented without XML mark-up.

5.6.1 *Tasks*

Profiles can be used to specify the preferences arising from tasks. A driver making deliveries might require a profile specifying that they are most interested in roads, then in house numbers nearby (by the combination of utility functions over null and distance) and somewhat interested in traffic conditions.

```
SPEC: deliveries
M: cMatch(task) = (value == delivery_driver)
U: utilityFn(road) = 1
utilityFn(traffic) = 0.7
utilityFn(house_nos) = 0.8
utilityFn(house_nos.distance) = strongPrefNearer(value)
```

We have also developed a tourist task which illustrates differing needs to those when on business. This includes a strong interest in tourist features, parks, etc., some interest in footpaths and a certain willingness to pay for data. Depending on the total cost of the data selected some may be omitted. Based on this specification footpath elements would be the first to be lost. Where different variants have different costs a degradation may be used rather than omission.

```
SPEC: tourist
M: cMatch(task) = (value == recreation:tourist)
G: goal(cost) = 400
U: utilityFn(tourist) = 0.89
utilityFn(land_use:park) = 0.8
utilityFn(footpath) = 0.5
```

A profile may be specialised for sub-classes of an aspect. For instance, a driver making deliveries is a specialisation of business, inheriting any functions it does not override. A general profile relating to business may be defined, which may form a base profile for many different work-related tasks. This profile may specify an express disinterest in tourist features (no variant of these elements will be selected even where goals do not limit the selection) and a price threshold which is greater than for a tourist.

```
SPEC: business
M: cMatch(task) = (value == business)
G: goal(cost) = 500
U: utilityFn(tourist) = 0
```

The difference between maps selected for tourist and business use is illustrated in Fig. 8. Note the business map does not indicate the nature reserve, castle, youth hostel, golf courses, footpaths, etc.

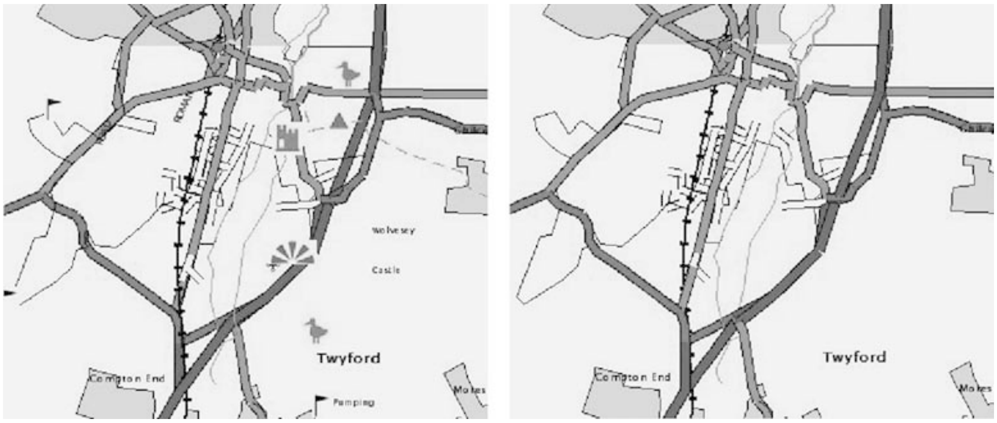
5.6.2 *Speed*

Here we describe preferences which may be due to driving a car at a moderate speed. As a result of the speed of travel we require the map to be downloaded within 25 seconds of the notification of the next predicted location. Due to vibration and distractions at this speed the screen density is limited in order to retain clarity in the display. The utility functions here differentiate between the different sub-types of road; for minor roads, according to their distance from the point of interest (here the centre of the map); and for a range of scales, as described earlier. Here, the preference will be for more general road information.

```
SPEC: car_moderate_speed
M: cMatch(speed) = (21 <= value <= 45)
   cMatch(transport) = (value == in_car)
G: goal(deadline) = 25
   goal(screen_density) = 2.2
U: utilityFn(road) = 1
   utilityFn(road:motorway) = 0.91
   utilityFn(road:minor) = 0.9
   utilityFn(road:minor.distance) = strongPrefNearer(value)
   utilityFn(road, map.scale) = mediumDetail(value)
```

Next, we illustrate a profile reflecting different preferences when moving slowly. The general utility for "road" is overridden for some sub-types here. Moving more slowly allows a longer deadline and a greater ability to read detail in the screen. The tradeoff between different types of road has now changed—at slower

Fig. 8 A comparison of maps for different tasks. Left-hand for tourist, right-hand for business. Both are 25km^2, density limit = 2.2 on a 400 × 400 pixel screen, 25s deadline over a 14kB/s network



speeds, minor roads nearby become more important than motorways. Similarly, the utility function over scale for roads now reflects a need for more detail.

```
SPEC: car_slow_speed
M: cMatch(speed) = (0 <= value <= 20)
   cMatch(transport) = (value == in_car)
G: goal(deadline) = 30
   goal(screen_density) = 2.5
U: utilityFn(road) = 1
   utilityFn(road:motorway) = 0.9
   utilityFn(road:minor) = 0.94
   utilityFn(road:minor.distance) = strongPrefNearer(value)
   utilityFn(road, map.scale) = highDetail(value)
```

Figure 1 illustrates maps for use at different speeds. The effect of utility functions on minor roads is visible, and the comparison with the unmediated map shows the exclusion of elements to meet time and screen density constraints.

5.6.3 Activity

We do not assume that the computer is the focus of all attention in ubiquitous computing. Profiles may reflect the impact of activities performed at the same time as using the system. One may define activities such as passenger or driver as specialisations of in-car navigation. An extract from the on-foot profile is shown below. The longer deadline and higher screen density allow the inclusion of far more detail. We also describe a stronger interest in paths and railway stations than when driving.

```
SPEC: on_foot
M: cMatch(transport) = (value == on_foot)
G: goal(deadline) = 90
   goal(screen_density) = 4
U: utilityFn(road) = 1
   utilityFn(building:railway_station) = 0.85
   utilityFn(path) = 1
```

5.6.4 Screen size

When the screen size changes the information shown for a given density may also change. Figure 9 illustrates how feature inclusion is affected by the density control as screen size changes. Note the removal of minor roads and major roads near the edge in the smaller screen's map.

The screen size of the device in use may also affect preferences over the granularity of data, as highly detailed map data would be lost on a low resolution screen, which might be encoded as below:

```
SPEC: small_screen
M: cMatch(screen_size) = (60 < value.x < 320)
   cMatch(screen_size) = (60 < value.y < 240)
U: utilityFn(*, map.scale) = generalDetail(value)
```

5.7 Combining specifications

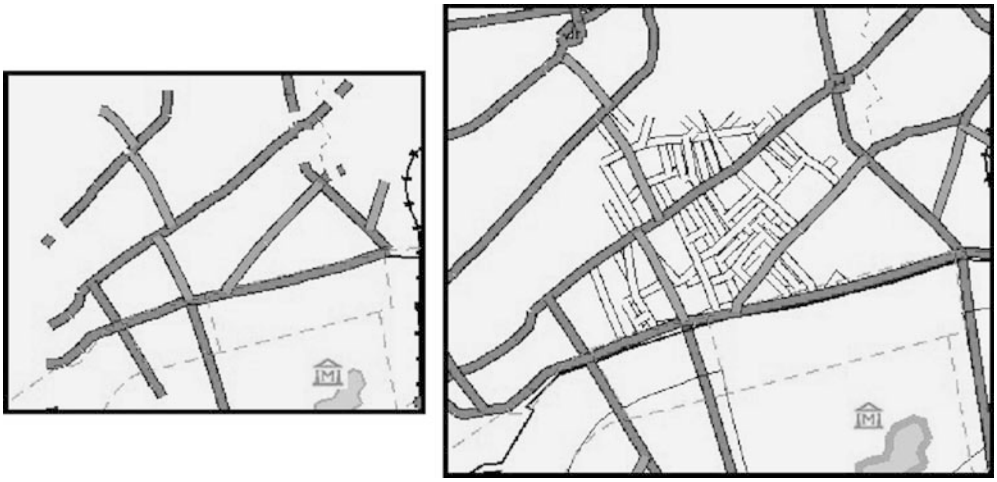
We have illustrated a range of profiles, enabled by different aspects of context. Given that we have several profiles which are relevant at any given time we require a technique to combine them. We take a combination of approaches for different parts of the profile.

5.7.1 Utility functions over elements

For elements a zero rating for an attribute is honoured, otherwise the best is taken. As the utility controls element inclusion and degradation order, the generous approach seemed most appropriate—a preference is to be honoured unless a clear, “absolutely unwanted” rating has been provided. Where utility functions for several different attributes (including null) exist for an element, they are combined as a product. Consider the following two specifications, both of which are enabled by their matching functions.

```
SPEC: car_moderate_speed
U: utilityFn(road) = 1
   utilityFn(road:minor) = 0.9
   utilityFn(house_nos) = 0.1
   utilityFn(road:minor.distance) = strongPrefNearer(value)
SPEC: deliveries
U: utilityFn(road) = 1
   utilityFn(house_nos) = 0.8
   utilityFn(house_nos.distance) = strongPrefNearer(value)
```

Fig. 9 Screen size/density effect. Left 300 × 300 pixel, right 400 × 400 pixel. Both are 4km^2, density limit = 2.50, deadline 30 s over a 3.6kB/s network



The combined utility functions for the types road, road:minor and house-nos are:

```
utilityFn(road) = 1
utilityFn(road:minor) = 0.9 * strongPrefNearer(distance)
utilityFn(house_nos) = 0.8 * strongPrefNear(distance)
```

5.7.2 Utility functions over variants

For variants, all the applicable functions for an attribute are combined by product. Any zero utility result will clearly override all other values. The lower value that a greater number of products (of values between 0 and 1) tends to give is not a problem as the utility is used to differentiate between variants realising one element. All variants will have the same functions applied to them, so the effect will be consistent. Where the variants have differing numbers of attributes some normalisation may be necessary. Consider the following two specifications, both of which are enabled by their matching functions:

```
SPEC: car_slow_speed
U: utilityFn(road, map.scale) = highDetail(value)
   utilityFn(*, map.scale) = medDetail(value)
SPEC: small_screen
U: utilityFn(*, map.scale) = medDetail(value)
```

The combined utility functions for the parameter map.scale on types road and * (all other types) are:

```
utilityFn(road, map.scale) = highDetail(value)
utilityFn(*, map.scale) = medDetail(value) * medDetail(value)
```

5.7.3 Goal functions

For goals, we take the lowest value for each resource and parameter. This will then reflect any limitations in time, cost, screen space, etc. which are being described. Where goal functions with different parameters are available for a resource we sum the minimum goals for each parameter. We give some feedback on the goals applied, for instance, a timing "progress" bar shows progress towards the deadline, advancing each second rather than with data loading. Consider the following two specifications, both of which are enabled by their matching functions:

```
SPEC: car_slow_speed
G: goal(deadline) = 30
   goal(deadline, map_area) = 0.2*value // in km²
   goal(deadline, speed) = -value/5     // in km/h
SPEC: battery_low
G: goal(deadline) = 20 // to reduce network use
```

The combined goal for the parameter deadline is:

```
goal(deadline) = 20 + (0.2*map_area) − (speed/5)
```

5.8 Notes on profile design

In developing our application and specifications and performing tests we have found various specifications which are not as desirable as they might initially seem. When we define utility functions taking distance from a point of interest as a parameter we use step functions, which has the effect of grouping elements of a type in concentric circles around the point of interest. A continuous function led to almost all elements taking a different utility. As we treat elements of the same utility together in adaptation this led to confusion over loss of information. Where elements of a type are grouped their inclusion or omission is clearer (particularly in the case of roads). Where each is treated separately absence (in reality) and omission (due to mediation) are harder to differentiate between in the final map, e.g., a small road in a dense network can be removed and go unnoticed until it is met in reality. A similar issue occurred when we considered using feature size (from their bounding box as described in metadata) to affect inclusion. While this may be sensible for some types of element it can also lead to spurious differentiation.

When using distance-related utility functions it is also important to consider the interaction between elements. Including minor and major roads at the centre of a map but not including major roads at the next distance out may not be desirable. Distance functions must consider the useful range of the type of elements, in the case of major roads the distance steps should be much larger than for minor roads.

Some features which are not immediately relevant are nonetheless useful navigation markers. Railways and rivers are useful features to know about when driving even if the driver has no intention of taking a train or boat.

We do not expect non-technical end users to write these profiles. Profiles are intended to be predefined and enabled automatically in response to sensed context and could be provided by specialists, from user studies, and a user might tune parameters. We expect that a base set of specifications would be devised as part of the application development or as standards used by multiple applications addressing the same domain. As new devices, sensed contexts and tasks emerge, new specifications may be added to support these. Similarly, we have not considered profiles which respond to user feedback. The acquisition of user reactions to maps was beyond the scope of our work and is of questionable applicability where the use of the device is not the user's main focus of attention. In any case, our profiles could be modified by user feedback if this was desired.

Automatic adaptation raises the issues of expectation and surprise for the user [20, 21]. If the system decided that the context had changed and made dramatic changes to the mediation specifications, as a result the user may find that the map is unexpectedly different in character from one request to the next. This needs to be treated as for any other user impacting behaviour: there is a need for predictability and/or transparency. We address this in five ways:

- Where the specifications reflect user context then a dramatic change in context could reasonably lead to a dramatic change in specifications—cause and effect are then perceived.

- Rather than having a specification for fast and another for slow a series of specifications reflecting small changes in speed would avoid dramatic changes in behaviour. By ensuring that the specifications reflect small changes in context as small changes in behaviour the impact of these small variations will be reduced.
- Some contexts or resources, such as network throughput, will vary slightly from request to request without any reason that the average user will perceive. Specifications need to provide sufficiently fine grained selection choices that minor resource fluctuations can be accommodated subtly. Appropriate use of caches may also be of benefit here.
- User interface tools may be used to prepare the user for changes and build expectations. For instance, changes in deadline can be indicated using a progress bar and changes in scale can be indicated by drawing a scale bar. Where the changed aspect of context is a system level issue, such as network characteristics then other indications (such as a coloured link quality icon) would be appropriate.
- Our application has a second window which provides information on the current context, enabled specifications, the network model, etc. A curious user, or one who has been surprised by an unexpected change in behaviour, could look at this interface to better understand changes in application behaviour. In some cases these interfaces also allow the user to override incorrectly sensed context values. The application also has controls for reloading data (and we plan to add one for loading more data by repeating the download deadline with existing metaddata) which should aid recovery from poor selections by the mediation system.

5.9 Applicability to other application domains

We have performed some initial experiments with Web pages and found that contextual mediation may be of benefit. However, it requires that the tasks being undertaken and semantics of the data are well understood. It is therefore most applicable as a tool over certain sites or where the Web is used as an interface to an application or pages containing well-defined types of data. Mediation may offer a framework for standard effects such as modifying text version and image selection depending on screen visibility, etc. We describe examples of contextual mediation of Web pages in [7, 15]. Its applicability to general Web browsing requires a sophisticated ontology and corresponding specifications which have been beyond our research scope so far.

Contextual mediation may also be applied to applications such as news alert services. The flow of interactions would be rather different, but similar techniques should be applicable. The semantic content of news stories may be used in providing utilities. Different utilities may lead to different levels of notification. Different representations of a news story may be applicable: depending on the level of interest, other activities been undertaken, the device being used, etc.

6 Experience

6.1 User studies

We performed an initial study into users' reactions to the maps which our system generated. The map application developed had some technical, HCI and data distribution limitations which would have made an extensive user test with the application difficult. We chose instead to take example maps which demonstrated the mediation being applied and provided them printed on cards to a small group of users (14 people). This enabled us to make a preliminary study of the reaction to maps at these sizes and compare reactions to mediated and unmediated maps.

The maps used in the survey were printed at 100dpi, which is approximately the pixel pitch of LCD screens used in the classes of devices represented. The card maps therefore had dimensions similar to those of a PDA screen. Control maps were produced where no mediation was performed. The control maps were saved as a JPEG with a compression set such that the image file took approximately the same size as the NTF data took when GZIPPed. The intention was to simulate a proxy responding only to system limitations: the image was delivered at the correct pixel size and the download time would be within the deadline. The effect of network delay and benefits of predictable delay could then be ignored in our test. The map data set was quite basic, for instance, contours, house numbers and traffic restrictions were not present.

We gave the subjects a series of three of the test maps: large and small mediated maps and a control map (either large or small). The context used in generating the maps was in-car, as a tourist at 30 mph. For each situation, they were asked to put themselves in the context of navigating in a car (the scale is similar to or larger than a road atlas, so the maps would be most appropriate for a passenger as the navigator) and to describe the route taken between marked points. Features used to identify places were to be described. The actual route taken was not noted, but the time taken to describe each was noted. A qualitative judgement of how easy the map was to use was also taken. The order varied with the control being first in half the cases, the small mediated map first in the other half. As the maps essentially took the form of road maps lack of familiarity was not considered to be a big problem although the trend was to speed up slightly by the third map. The change of order was designed to counter this effect.

The first and most striking result of the timings was the variation in times taken to describe routes—the range of times was five to 64 seconds. All subjects were consistently fast or slow in broad terms. To make

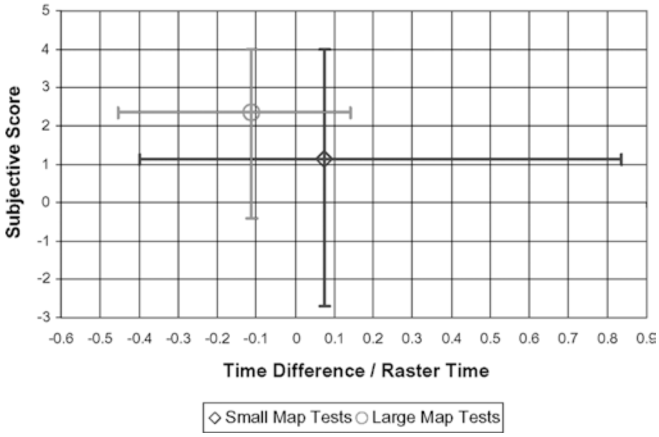


Fig. 10 The ratings from the user tests

general comparisons the timings needed normalisation. We took the difference between the mediated map and the unmediated map and divided this by the unmediated map time. This then gives the proportion of the time taken by which the mediated map is faster or slower to use. We present the mean, 10th and 90th percentiles in Fig. 10 on the horizontal axis. A negative value indicates that the mediated map was faster. The vertical axis of Fig. 10 gives a qualitative score, calculated from the response to the subjective questions. A higher score indicates a subjective preference for the mediated map over the unmediated map.

The horizontal axis of Fig. 10 shows us that the large mediated map was faster (11%) to use on average than the unmediated map, while the small mediated map was slightly slower (7%). In fact, in both cases nine of the 14 subjects recorded faster times with the mediated map. In the case of the small map the high mean and 90th percentile comes from three subjects who were much slower. The range of absolute times for the large map was also substantially less than for the small map, reflecting a general preference for the larger map. In cases where the mediated map was found to be slower it was slower by a smaller degree for the large map. The speed difference was not substantial for either the small or large mediated map over the unmediated map. We believe that the real benefits in speed of use would come as the unmediated map became more cluttered.

The subjective scores (on the vertical axis) indicate that in general users prefer the mediated map (10th percentile for the large map has a score of −0.4, where 0 shows no preference), which at worst is similar to the unmediated one and at best is less cluttered.

In summary, for most users the mediated maps were both faster to use and subjectively better. We believe that if the map data offered was richer then the benefits of mediation would be greater and felt by more of the users.

Comments by test subjects and observations of their response to the maps were recorded during tests and provide useful feedback into the design of the application and specifications. Two subjects noted that they took a different route for the mediated and control maps where minor roads were obscured by other features in the control map. One assumed that roads would continue where obscured and used these in their route. This is a clear demonstration of the potential benefit of mediation—the most important data can be revealed and extra information only included where the user has sufficient interest and the context of use make its display appropriate.

It was felt that a route plan would provide a more accurate representation of interest due to distance than a point of interest, i.e., describing a path of interest and utilities radiating from this, whereas we simply used a point at the centre of the map at this stage in prototyping. In the larger area map some subjects still found it cluttered and would have zoomed if using the actual application. The drawing of the map was not ideal: lack of road names was cited as a problem by several subjects. While these subjects tended to mark all the maps down for the deficiencies in presentation these issues can be rectified with further development of the application and map rendering. One subject commented on the poorer ‘‘quality of drawing’’ in the control map. The image degradation was not substantial for most of the maps, so it is not surprising that few subjects responded consciously to the lower quality of the graphics. The impact of JPEG compression would be greater had the maps contained more features, particularly small text giving road names, etc.

These initial user studies showed that the mediated maps are generally found to be at least as good as those without mediation, but the full power has yet to be realised. The application currently lacks the refinement which would have made a more substantial user test a useful proposition. Where the range of data which might be displayed is wider then the benefits are likely to become greater as the control map would become harder to read. Where the level of user interaction is greater then user control over the map displayed is possible, such as zooming and modifying goals, which we expect to enhance usability.

6.2 Meeting deadlines

In our application a model of network throughput and latency to the various sources is maintained, and network throughput may be reserved. While this reservation may not be end-to-end, or provide guarantees (and will certainly only be an estimate in the case of wireless networks), we have found that this provides a sufficient limitation to meet (or nearly meet) download time goals and reduce variance in loading time. We have verified this using simulated lossy links with throughputs relevant to wireless networks (9.6k to 1Mb/s), including under a range of simulated loss rates (0 to 10%), and deadline ranges (5 to 60s).

Figure 11 presents download times (the mean, 10th and 90th percentile) for mediated map data with a range

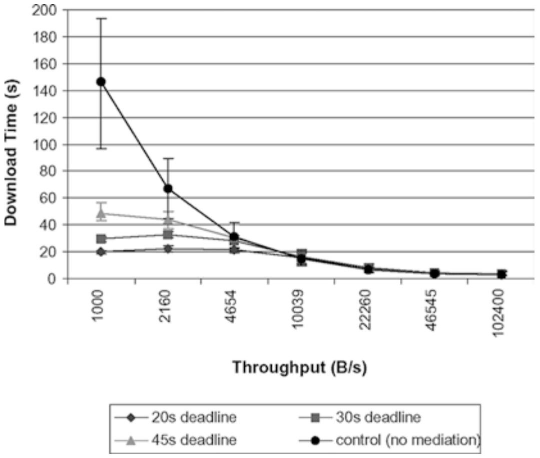


Fig. 11 A comparison of mediated and unmediated download times

of deadlines and compares them to the unmediated case. The application displayed a map for an area of 10.7km^2 (based on a 3 × 3 km area of interest expanded to fit the window dimensions). The times taken would vary if the map area or feature density were changed, but the overall pattern and validity remain. It can be seen that the network model was not perfect, and the deadlines were not always being met. However, at lower throughputs the maps are loaded in a time close to the deadline while the control case takes substantially longer. At higher throughputs the traces converge as the data fails to consume all the resources available. In all cases the mediated application shows much less *variation* in the download time between requests than the unmediated case, so it provides a more consistent user experience. We shall not describe these tests further here, as they reflect the quality of the network monitor (which is not the focus of this paper) rather than contextual mediation as a tool—see [15] for a detailed presentation.

The inability of unmediated delivery to meet deadlines is greatest when the maps are largest and so the volume of data greatest. In these cases, the metadata overhead of the mediated solution is greatest. However, even where a substantial benefit in time is seen a usable map can be provided. Fig. 1 illustrated this case for a 9km^2 map of an area of London loaded over a 3.6kB/s network, where the unmediated map took 46s (without metadata) and the mediated map took 19s (including metadata) against a deadline of 25s. In Fig. 12 we illustrate this again, for a 100km^2 area of rural southern England and a 20s deadline over the same network. The unmediated map took 46s to load, while the mediated map took just 7.4s. In rural areas the map area required in order to achieve a failure to meet a deadline is much larger. However, the map is intended for use in navigating at 50 mph, so the greater area would also be a practical requirement. The mediated map remains a practical tool—and shows that the ability to meet deadlines gives a usable map not just a theoretical result.

7 Related work

7.1 Use of context

We identify six uses of contextual information, drawing from [4, 15, 22, 23]:

a. *Contextual sensing*: where the context is sensed, and information describing the current context, e.g., location and temperature, can be presented to the user.
b. *Contextual augmentation*: where context is associated with data. For example, records of objects surveyed can be associated with location, meeting notes can be associated with people in the meeting and the place the meeting was held [23].
c. *Contextual resource discovery*: e.g., to cause printing to be on the nearest printer.

Contextual adaptation [22] is used in the literature to describe the following three cases. We draw a distinction between these cases: where context causes an action, where it is involved in modifying an action which has been caused separately, and where it is involved in adapting the presentation of an application or data.

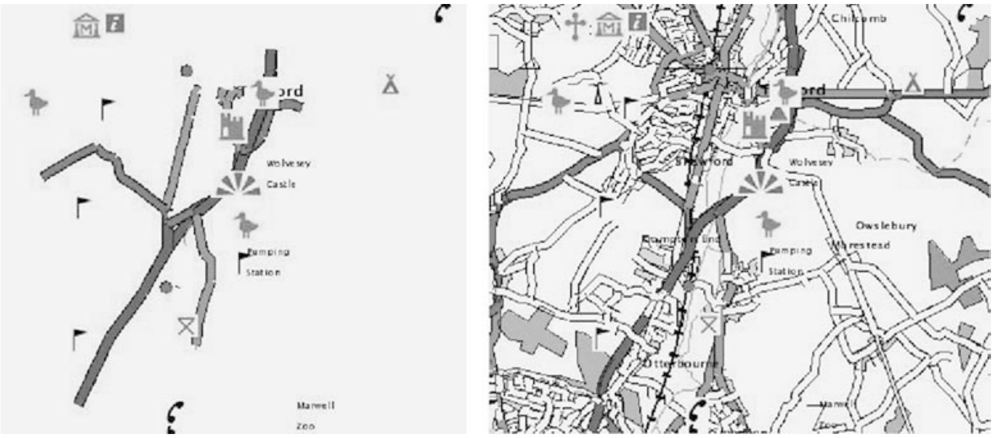


Fig. 12 Meeting deadlines. Left-hand mediated, took 7.4s; right-hand unmediated, took 46.1s. Both are 100km^2, driving at 50mph as a tourist, density limit = 1.60 on a 400 × 400 pixel screen, deadline 20s over a 3.6kB/s network

d. *Context triggered actions* [4] such as loading map data for the next location predicted, exchanging business cards, etc. [19]. In our map-based application we make use of changes in location to trigger data loading.
e. *Contextual mediation* is the use of context to modify services provided or the data requested to best meet the needs of the user arising from the context of the interaction, while working within the limitations of the context [15].
f. *Context aware presentation* refers to the adaptation of the user interface or the presentation of data. This is closely related to contextual mediation and the two are expected to work together, e.g., to load the data which is most appropriate (mediation) for the best rendering (presentation). We do not consider this is in detail here, but expect to address some of these issues in the future. Some presentation adaptation techniques are discussed in [24, 25].

7.2 Mediation

We believe there is a need for a more flexible solution to the problem of context-sensitive data provision than the fixed use of standard translations or restrictions according to device parameters, as described in much of the literature, e.g., [13, 26, 27, 28, 29, 30]. Our approach has the advantage that use is not tied to particular classes of devices and does not require separate effort to tailor maps to specific devices. Being more general, this approach facilitates the use of emergent devices, while allowing tailored data to be used where the effort has been made to provide it. These specialised versions may be part of the solution, particularly for very limited devices. However, when using devices of the next level of capability (PDA to laptop, kiosks, in-car systems, etc.), the device itself ceases to be an absolute limitation. The user interface and network connection are often both sufficient to support more than the most basic data presentation, although still being limited in comparison to desktop PCs. Maintaining context-sensitive profiles, which support any specific data generated, can better support a device and user-centric adaptation of the data presented.

Our approach of selecting data according to its properties has similarities to the approach in [31], where they describe an interplay of willingness to degrade data against urgency. Their approach assumes that the quality of the data can be inferred from its size, which is not true for all data.

As described in the Internet Engineering Task Force (IETF) RFC 2295 [27], the hypertext transport protocol (HTTP) supports negotiation among variants for a given uniform resource identifier (URI). In this system a negotiable URI has a list of variants, whose attributes are described. This mechanism meets the basic need for selection amongst variants of media. However, there are some areas in which it does not support the flexibility we seek. The negotiation is on a per-element basis. Any requirement to negotiate towards a deadline or to ensure the full presentation fits within a screen size or treat other resource limits cannot be met where a negotiation over elements does not consider their cumulative effect, or tradeoffs between elements. Secondly, the description of variants does not include any information about the semantics of the element (although issues regarding this are starting to be captured in work by the World-Wide-Web Consortium (W3C) [9]).

The W3C Composite Capability Preferences Profile (CC/PP) [29] notes that mechanisms such as accept headers (in HTTP) and “ALT” tags (in HTML) are somewhat limited and described the application of the resource description framework (RDF) [32] to describe user preferences and device capabilities in a general content negotiation solution. The IETF CONtent NEGotiation (CONNEG) group’s work, described in RFC 2533 [28], gives a system for describing preferences for media based on attributes of the media. The CONNEG group has addressed similar issues to the CC/PP group and there are efforts to ensure that the two systems can inter-operate. They include support for a detailed description of the hardware platform, software capabilities and preferences for languages and security. The encoding does indeed address the difficulties with the HTTP negotiation with regard to the richness of preferences which may be specified. While this addresses the need for transmittable preferences we shall have to look elsewhere for techniques to describe the data and perform the selection. Also, there is no mechanism described for relating these specifications to context.

A proxy-based map adaptation client designed for use with hand-held devices is described in [33]. Maps are adapted by feature elimination due to (semantic) element type, feature size and by eliminating detail from features. These ideas are similar to the basis of those presented in the variants we use. The adaptations performed are designed to reduce network load for slow links and to reduce clutter on small screens. The adaptations are performed at a proxy, according to given user preferences for element types and a stated feature size / screen size metric. In the developed system no means for the context or user to select preferences was given and hard-coded values were used. While a demonstration of the possibilities for reducing data volume by feature (GIS layer) elimination and/or coarsening of detail is given, the possibility of meeting deadlines is not discussed and network testing is left as future work. The adaptation due to screen size simply filters small features out. In many ways their approach is specific to maps and provides a coarser specification of adaptation than our system. In filtering small features the treatment applied to point features (which may be rendered as more than single pixels) is unclear. Consideration of feature density on the screen is noted as possible future work.

From a tourism perspective Zipf [34] considers various ways in which maps may be adapted to meet different needs, in terms of device, user task and user culture. They examine issues of layout and presentation, including the use of selective presentation to aid the user's focus, as well as possibilities for modifying the data. However, little information is available beyond the ideas of techniques which may be useful. As a general framework it is informative and supports our ideas, but offers little guidance from experience.

In [35] a different approach to drawing maps is discussed. The problem of lack of screen space and confusingly cluttered displays is addressed by omitting large amounts of data and using maps without constant scale, emphasising certain features. Their approach is aimed at maps to describe routes for driving and so they omit all features away from the route. The creation of maps to fit small (e.g., PDA) screens is described and user tests are largely positive. Problems identified with this approach include the difficulty of recovering from errors and the need for detail around the destination to find parking places, etc. Our approach contrasts with this by using a sequence of maps, adjusting to speed and position along the route by showing different levels of detail. Our approach is also suitable for a browsing style of map reading as well as route navigation.

The needs of pedestrians in navigation systems are given consideration in [36]. In this case, the accuracy of sensed context may well be worse than in cars and the tasks engaged in may be more complex. Issues addressed include adaptation to varying location precision, adaptation to changes in speed and limiting the cognitive load arising from the directions or map given. The different elements of their presentation are broadcast with different frequency, which depends on their weight, building presentations in stages. Apart from this adaptation is a matter of rendering. The weights are assigned to the elements rather than in response to the context and presentations seem to require well-controlled authoring.

Negotiation to select the most appropriate PDF documents according to context is described in [30]. Their approach centres on selecting the best document from a set, rather than combining elements, according to device capabilities and user preferences. While the model supports a range of adaptation the context aspects responded to are limited to the device, network and user—lacking the fine-grained response of our approach to different influences and highly structured data.

In examining techniques for route planning [37] considers a Markov decision process-based technique which might also be applicable to our selection process. It considers a combination of weighted goals which may be achieved by some series of steps, each having some cost and some benefit. The ability to plan routes which cope with likely errors is also interesting if applied to accommodating network problems or other sudden context changes.

In summary, our framework supports a rich description of both data and response to context in selecting data, a consideration of data structure, a reaction to many kinds of resource limitation and flexibility in the relationship between perceived quality and data parameters which do not appear in the literature.

7.3 Media adaptation and rating

Adaptation will generally result in the loss of information from the original version, e.g., making an image smaller, reducing the colour depth and omitting data. Mediation should be seen as being the process of moving to the best point within a space rather than purely as degradation, although much of the literature, such as [16, 26, 38], refers to it in these terms. A high colour, high resolution image may not offer any benefit over a version which has lost much of this information where the display cannot convey the extra detail. In many cases, presenting all offered data is likely to be overwhelming and contain much irrelevant information, so omission does not constitute a degradation, from the user's perspective.

The adaptation techniques that may be applied can include [31]:

- Information abstraction, e.g., lossy image compression, text summarisation.
- Modality transformation, e.g., video to still image sequence, audio to text.
- Data transcoding, e.g., format conversion, often combined with abstraction.
- Selection, which may be by data prioritisation, or binary selection.

8 Conclusions

In this paper we have given a classification of the uses of context in applications. We have described and demonstrated a system of profiles to enable *contextual mediation* to support users of semantically rich data-intensive applications on a wide range of devices, using a wide range of networks and engaged in a wide range of tasks and activities. Our approach is a general one which can make use of specific adaptations without relying on them.

Data forming *documents* is described in metadata, which describes semantics, syntactic properties and the structure of data. The selection space formed by structured documents is treated through *utility functions* over *semantic type*; other attributes of abstract media *elements*; and over attributes of *variants* which realise elements. The mediation is subject to limits from *goal functions* describing limits on managed *resources*. The treatment of the semantics of the data in selecting functions offers a granularity of response to the user's needs, which earlier approaches, such as uniform scaling of data, could not achieve.

Combinations of these descriptions of preference and need are selected according to context. *Matching* *functions over context aspects* allow preferences which reflect a narrow set of needs to be defined and selected separately to those reflecting other needs. As the detected aspects of context and needs due to context evolve, the profiles can be managed in a modular fashion. These context-selected profiles enable a wide range of differentiation in semantically-rich applications with well-structured data. Several profiles may satisfy their context matches at one time. A mechanism to combine these profiles was also described.

We have described mediation limited by download deadlines and a network throughput and latency model. The network model is sufficiently accurate that deadlines are met in most cases, thus providing a more predictable user experience. We have also implemented a screen and data density model, which limits clutter on small or otherwise hard to read screens. User tests based on our prototype application and basic map data indicates that mediated maps are useful and generally at least as good as those presented without mediation.

Overall, we believe that contextual mediation offers a general mechanism to support many applications and contexts, while offering a more powerful and sensitive adaptation to context than those in the prior work.

8.1 Future work

We are continuing to refine our application, the specification of profiles, context sensing and resource management. The adaptation of map data described is essentially a base case. The power of mediation should become stronger as the range of data (including hyperlinks and multimedia) becomes greater and the potential for resource overload increases. Applications of the concepts to other applications is also being investigated. The context-aware presentation of data is also an area of interest, such as modifying line widths, text sizes, etc. This would complement and interact with contextual mediation, particularly with respect to screen space management. As noted during our user tests, the extension of our location model to include paths of interest would also be useful.

Acknowledgements This work was funded by the EPSRC grants GR/M 26428 "LinkMe" and GR/R 95715 "Aedus". Map data was generously made available by Ordnance Survey. Feedback and suggestions from Arosha Bandara, Danius Michaelides, Les Carr, Wendy Hall, David De Roure and the reviewers are gratefully acknowledged.

References

1. Weiser M (1993) Some computer science issues in ubiquitous computing. Comm ACM 36:75–85
2. Chalmers D, Sloman MS (1999) A survey of quality of service in mobile computing environments. IEEE Comm Surv. http://www.comsoc.org/pubs/surveys/. Cited 2 February 1999
3. Katz RH (1995) Adaptation and mobility in wireless information systems. IEEE Pers Comm 1(1):6–17
4. Schilit B, Adams N and Want R (1994) Context-aware computing applications. In: Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, December 1994
5. Chalmers D, Sloman M and Dulay N (2001) Map adaptation for users of mobile systems. In: Proceedings of the 10th International World Wide Web Conference (WWW-10), Hong Kong, China, May 2001
6. Cartwright W, Peterson, MP and Gartner G (1999) (eds) Multimedia cartography. Springer, Berlin Heidelberg New York
7. Chalmers D, Dulay N and Sloman M (2004) Metadata to support context aware mobile applications. In: Proceedings of the 5th IEEE International Conference on Mobile Data Management (MDM 2004), Berkeley, CA, January 2004
8. Kerhervé B, Pons A, v. Bochmann G and Hafid A (1996) Metadata modelling for quality of service management in distributed multimedia systems. In: Proceedings of the 1st IEEE Metadata Conference, Silver Spring, MD, 16–18 April 1996
9. Berners-Lee T (1998) Semantic Web roadmap. http://www.w3.org/2001/sw/. Cited September 1998
10. Lee C, Lehoczky J, Rajkumar R and Siewiorek D (1999) On quality of service optimization with discrete QoS options. In: Proceedings of the IEEE Real Time Technology and Applications Symposium (RTAS'99), Vancouver, British Columbia, Canada, June 1999
11. Walpole J, Krasic C, Liu L, Maier D, Pu C, McNamee D and Steere D (1999) Quality of service semantics for multimedia database systems. In: Proceedings of Data Semantics (DS-8), Rotorua, New Zealand, January 1999
12. Noble BD (2000) System support for mobile, adaptive applications. IEEE Pers Comm 2:44–49
13. Mohan R, Smith JR and Li C-S (1999) Adapting multimedia Internet content for universal access. IEEE Trans Multi 1:104–114
14. Watson A, Sasse MA (1997) Multimedia conferencing via multicast: determining the quality of service required by the end user. In: Proceedings of the International Workshop on Audio-Visual Services over Packet Networks (AVSPN '97), Aberdeen, UK, September 1997
15. Chalmers D (2002) Contextual mediation to support ubiquitous computing. PhD thesis, Imperial College London http://www.doc.lc.ac.uk/~dc/pubs.html
16. Noble BD, Satyanarayanan M, Narayanan D, Tilton JE, Flinn J and Walker KR (1997) Agile application-aware adaptation for mobility. In: Proceedings of the 16th Symposium on Operating System Principles (SOSP-16), Saint-Malo, France, 5–8 October 1997
17. Kim M, Noble B (2001) Mobile network estimation. In: Proceedings of the 7th International Conference on Mobile Computing and Networking (MobiCom 2001), Rome, Italy, July 2001
18. Gecsei J (1997) Adaptation in distributed multimedia systems. IEEE Multi 4(2):58–66
19. Schmidt A, Aidoo KA, Takaluoma A, Tuomela U, van Laerhoven K and van de Velde W (1999) Advanced interaction in context. In: Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing (HUC'99), Karlsruhe, Germany, June 1999
20. Thimbleby H (1990) User interface design. Addison-Wesley, Reading, MA
21. Höök K, Karlgren J, Waern A, Dahlbäck N, Jansson CG, Karlgren K and Lemaire B (1996) A glass box approach to adaptive hypermedia. J User Model User Adapt Inter 6(2-3):157-184
22. Dey AK, Abowd GD (2000) Towards a better understanding of context and context-awareness. In: Proceedings of the Workshop on the What, Who, Where, When, and How of Context-Awareness, Conference on Human Factors in Computer Systems (CHI2000), 1–6 August, The Hague, The Netherlands

23. Brown PJ, Bovey JD and Chen X (1997) Context-aware applications: from the laboratory to the marketplace. IEEE Pers Comm 4(5):58–64
24. Marttila O, Vuorimaa P (2000) XML-based mobile services. In: Proceedings of the 8th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG 2000) Plzen-Bory, Czech Republic, February 2000
25. Schmidt A (2000) Implicit human computer interaction through context. Pers Technol 4:191–199
26. Fox A, Gribble SD, Chawathe Y and Brewer EA (1998) Adapting to network and client variation using infrastructural proxies: lessons and perspectives. IEEE Pers Comm 5:10–19
27. Holtman K, and Mutz A (1998) Transparent content negotiation in HTTP. IETF Network Working Group, RFC 2295, 1998
28. Klyne G (1999) A syntax for describing media feature sets. IETF RFC 2533, 1999
29. Reynolds F, Hjelm J, Dawkins S and Singhal S (eds) (1999) Composite capability/preference profiles (CC/PP): a user side framework for content negotiation. W3C NOTE-CCPP-19990727, July 1999
30. Lum WY, Lau FCM (2002) A context-aware decision engine for content adaptation. IEEE Perv Comput 1(3):41–49
31. Ma W-Y, Bedner I, Chang G, Kuchinsky A and Zhang HJ (2000) A framework for adaptive content delivery in heterogeneous network environments. In: Proceedings of the Multimedia Computing and Networking Conference (MMCN00), San Jose, California, 24–26 January 2000
32. Lassila O, Swick RR (eds) (1999) Resource description framework (RDF) model and syntax specification. W3C Recommendation W3C-rdf-syntax-19990222 (W3C 1999)
33. Abdelsalam WAEF (2001) Maintaining quality of service for adaptive mobile map clients. Master's Thesis, University of Waterloo
34. Zipf A (2002) User-adaptive maps for location-based services (LBS) for tourism. In: Proceedings of the International Congress on Tourism and Communications Technologies in Tourism (ENTER 2002) Innsbruck, Austria. Springer, Berlin Heidelberg New York
35. Agrawala M, Stolte C (2001) Rendering effective route maps: improving usability through generalization. In: Proceedings of ACM SIGGRAPH 2001, Los Angeles, CA, 12–17 August 2001
36. Baus J, Krüger A and Wahlster W (2002) A resource-adaptive mobile navigation system. In: Proceedings of the International Conference on Intelligent User Interfaces (IUI02), San Francisco, CA, January 2002
37. Bohnenberger T, Jameson A (2001) When policies are better than plans: decision-theoretic planning of recommendation sequences. In: Proceedings of the International Conference on Intelligent User Interfaces (IUI2001), Sante Fe, NM, 14–17 January 2001
38. Gerfelder N, Jung H, Santos LM and Belz C (1998) Challenges to deliver 2D/3D content for multimedia applications in mobile environments. Demon Fut Wirel Info Infra 1:82–89