Submission to CaberNet Workshop, October 2001
Research theme: applications for heterogeneous platforms / mobile computing

Context-Based Specifications for Data Mediation to Support Mobile Systems

Dan Chalmers, Naranker Dulay, Morris Sloman
Department of Computing, Imperial College, 180 Queen's Gate, London SW7 2BZ, U.K.
`e-mail:{dc,nd,mss}@doc.ic.ac.uk`

Abstract Application level adaptation is widely cited as a practical response to the variation in context which mobile and ubiquitous computing are subject to. One approach to adaptability is to mediate the data downloaded and presented to the user. This may be achieved by selecting a sub-set of the possible data, and by selecting amongst different representations of the information available. We present a specification technique in order to describe user preferences due to context, for selecting amongst offered data. The specifications controlling which data to download are given in terms of the semantic types and attributes of the data, and goals such as deadlines. A mechanism for associating specifications with contextual conditions, and combining these specifications is outlined. This enables specifications which reflect aspects of context to be written, managed and applied in an effective way.

1 Introduction

Mobile and ubiquitous computing devices exhibit a wide range of user interface devices, a variety of network connections subject to various environmental effects, and these devices may be applied to a wider range of tasks than traditional computing systems [1,2,3,4]. The deployment of applications on mobile or embedded devices exposes the user to this range of variation in behaviour, which many of today's applications are not able to accommodate gracefully.

There is a need to provide for context awareness in mobile and ubiquitous computing to manage the incompatibility and frustration which can arise from the variations in device capabilities and user preferences. The wide range of variation renders many of the low-level network approaches to QoS management inappropriate or infeasible. Application level adaptation seems, then, to be appropriate.

Data, such as maps, has some content which is of general interest, and other content which is more specialised. When navigating in a car, details of roads, traffic, etc., are important. A hiker, on the other hand, may consider that footpaths, hills and rivers are more significant. Delivery drivers and emergency services will probably know the general geography of an area, but may want to identify detail such as house numbers and temporary diversions. Other information, such as administrative boundaries, spot heights, etc., which might clutter the map on a small screen, would be undesirable.

While most map information is static in nature, feature updates can be propagated by on-demand delivery in a more transparent manner than for statically stored maps, such as on CD. The potential volume of map data is also high, making static storage on limited devices a problem. Highly dynamic information, such as weather forecasts, traffic conditions or local entertainment information could be provided over the geographical features through overlays or hyperlinking. Just-in-time delivery of this information is usually preferable. However, the wide range of data may easily lead to information overload: for the display, network bandwidth, or for the user's ability to digest it.

A selection mechanism must therefore be able to differentiate between necessary, interesting and unnecessary data (and many degrees in between). This must be balanced against the need for timely and predictable delivery delay, working within screen space and cost limits, etc. Different users, in different situations, will have widely varying preferences. The user is best served by data which is finely adjusted to their needs. The range of device capability, and tasks users apply them to will be expanding for some time. Therefore it is undesirable for content providers to be obliged to provide 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 content negotiation which allows for general specification of requirements, and predictable and unobtrusive adaptation according to context is needed. This paper describes a system for describing specifications due to context, and combining specifications arising from different aspects of context, such as device, task, social situation, etc.

We are working with map-based applications, but our techniques are intended, and implemented, to be generally applicable to many other applications such as web browsing from mobile devices. The standard vector map formats, which we are using, provide data with a rich structure typical of many emerging media standards, and so the selection techniques are applicable over other media. A fuller account of the process of selection, our implementation, and initial results is given in [5].

2 Specfications for Data Selection

2.1 Selecting Amongst Structured Data

As discussed, there is a need for users to select a subset of broadly relevant and technically acceptable information to be displayed. For example, current solutions for web based applications fall into three broad categories:

- Specific data formats are unconditionally de-selected by the user, e.g., the "no images" selection commonly found on web browsers.
- Weightings are defined for specific formats (e.g., JPEG = 0.9, GIF = 0.8) or text language (e.g., French = 0.8, English = 0.5) to indicate the user's preference or capa-

bility of the user's device. These weights are often used to control server side selection, e.g. [6,7].

- Standard transcoding is applied to data in order to provide it in a particular format, or to meet device restrictions, such as image size or colour depth. Transcoding is generally offered to meet the needs of a small range of target devices [8].

These techniques suffer from significant limitations, particularly for the class of applications we are studying. Where structured data are being used, selection according to the data's format is unlikely to fully capture the user's needs. In web browsing, it may be sufficient to say that one dislikes images. However, where all data are in one format selection based on format is not meaningful. In this case it is desirable to be able to express preferences related to the *semantics* of the information, e.g., display roads but not contour lines.

We address this issue through the use of *elements*. An *element* is a generalised part of the data, which fulfils a specific role in a document. In a map this may correspond to a *feature*, e.g., the M1 motorway, or River Thames. Elements are instances of a *type,* which defines its semantic content, e.g., road, river or building. Motorways, major-roads, and minor-roads are all sub-types of road. The Dublin Core meta data [9] also uses the term "type" to describe a similar concept.

In our approach an element is represented by one or more *variants*, which may be provided by hand or automatically. For instance, a map may contain representations of the M1 motorway surveyed at 1:1000, 1:50000 and 1:100000, relating to lesser inclusion of fine detail. Similarly, a picture can have variants relating to different resolution images. A variant is described by *attributes*, which may be general, such as size (in bytes), or format specific, such as survey scale. Variants are units of data which may be individually requested.

Our approach to selecting data according to its properties has similarities to the approach in [10]. They describe an interplay of willingness to degrade data against urgency. We do not regard the selection process as being solely about degrading data: The total data about a geographic area is likely to be overwhelming and contain much irrelevant information, in which case omission does not constitute a degradation, from the user's perspective.

2.2 Specification of User Preferences

2.2.1 Weights

Users specify their preference for elements of a particular type by associating a *weight* with the type, for instance, to describe a preference for displaying information about roads rather than rivers. The types are structured through specialisation, and one weight definition may supersede another where one considers a general type, and another a more specific type, e.g. a weight for "motorway" would be applied rather than a weight for "road".

2.2.2 Utility Functions

Preferences for different variant representations of an element are defined using *utility functions* applied to the attributes of the variants. For instance, for in-car navigation various features are relevant. For roads, one might define utility functions relating to scale reflecting the fact that a lot of detail makes the map harder to read, while a large scale is often still useful. For hikers, there might be less of a tolerance for the loss of detail of bigger scales, but more time to read and use the greater detail of the smaller scales. We measure utility across the user's perception, rather than in relation to some ideal or original version, such as in [11].

The variant may have a utility function applied to each attribute described (in meta data). The utility values for the various attributes are combined to derive an *overall utility*, for the variant. As each attribute's utility value, has been normalised to between zero and one, we simply multiply all utility values together for each variant. The use of a product function has the benefit that any variant which is unacceptable in some aspect (zero utility) registers as a wholly unacceptable variant.

2.2.3 Selecting Utility Functions

It should be clear that these utility functions do not say anything about the perceived importance of roads within a map. If one were navigating it is likely that the "road" data would have a high weighting. However, a tourist might be less interested in the detail of the road than in historical monuments and restaurants, so loading unnecessary detail would delay the loading of more important information, and clutter the display. Utility functions over attributes can aid in describing these preferences. When defining user preferences it is important to be clear whether one is describing the importance of a feature (type weighting) or the effect of the representation quality on the perception of the data (utility function over attributes of a variant). The *types* of data a variant represents may be used to select amongst different utility functions to be applied to each attribute, to reflect different ways in which the perceived utility of the data varies.

2.2.4 Goals

The final part of the specification is rather simpler. This part describes goals to be applied over the whole document retrieval. To date the only goal we have worked with is a download deadline. Where bandwidth is limited with respect to the download deadline this has the effect of constraining the variants selected. Cost might also be a universally applicable constraint to work within

Where bandwidth is not highly constrained, with respect to the data volume and deadlines set, then satisfying the goal will be trivial. However, in many mobile or embedded devices resources such as screen space will continue to be limited due to the practicalities of device form factor. This is related to, but different from, the use of utility functions to describe a preference for images which fit within the screen.

2.3 Selection

We will not discuss selection mechanisms in detail here. However, we assume that a meta data description can be queried from the data sources, before requesting the actual data. The selection process follows an algorithm which treats differing types of data according to their weights, and selects amongst the variants of each element according to their utilities. The algorithm is specified to maximise the perceived utility to the user, within the resources available due to the goals. In addition to meta data, we assume that near the device making the selection there is some model of available resources which the goals may be applied to. In the case of deadline, we assume that a model of network bandwidth and round trip time to the various sources is maintained, and that network bandwidth may be reserved, locally at least.

Fig. 1. Example of Selection in a Map

We have found that this provides sufficient limitation to provide a reasonable restriction of download time and variance. We have verified this using simulated slow and lossy links with bandwidths relevant to the ubicomp domain (9.6kb/s to 1Mb/s), including under a range of simulated loss rates (0 to 10%), and deadline ranges (5s to 60s).

An illustration of a sample map, taken from our prototype application, is shown in figure 1 (map data copyright Ordnance Survey, 1996). In this case there is only one variant offered for each element. The second map illustrates sufficient features dropped to save 54% of the data volume required to describe the lower map.

3 Building Context Driven Specifications

Various parts of the specification required at any time will be due to the many factors associated with context, such as location, co-location, social situation, task, screen size, etc. Each of these *context aspects* will have some effect on some part of the overall specification. However there may be a large number of combinations of these context aspects which may be met by any given user, e.g., a user may use several devices each day, and undertake various tasks. Clearly anticipating all combinations of context aspect which might be met, and defining specifications tailored for each, is not practical. Rather, we seek to describe the specifications due to each of these aspects of the context. So, a single specification would be written for each device commonly used, e.g., desktop, laptop, or PDA. Another specification for each task engaged in, e.g. various jobs at work, shopping, commuting, etc., another for co-location with groups of people, e.g., colleagues, customers, in-public, etc.

The issues involved in sensing context shall not be addressed in detail here. However, we do assume that context may be sensed in a variety of ways, providing a multi-faceted description of context, as in [12]. We identify five uses of contextual information, drawing from [2,13]:

a. "Contextual sensing" – where the context is sensed, and information describing the current context, e.g. location, temperature, can be presented to the user.

b. To associate context with data, known as "contextual augmentation". For example records of objects surveyed can associated with location, meeting notes can be associated with people in the meeting and the place the meeting was held [14].

c. To enable "contextual resource discovery", e.g., to cause printing to be on the nearest printer.

   "Contextual adaptation" [13] is used to describe both of the following two cases. We draw a distinction between the case where context causes an action, and where it is involved in modifying an action which have been caused separately.

d. "Context triggered actions" [2] to trigger actions such as loading map data for an area to be entered. In our map based application we make some use of changes in location to trigger data loading, and expected arrival time to limit download time.

e. "Contextual mediation" – using context to modify a service based on context. The focus of our specifications is on the use of context to mediate interaction – to describe limits and preferences over a large range of data. These specifications will be selected according to, and reflect, a wide range of contextual input.

We wish to associate a specification with certain conditions in one or more aspects of context. A specification will be applicable when one or more context aspects have certain values, or are within or beyond some value. We use a set of matching functions, from a subset of those aspects in the description of the total (known) prevailing context, to achieve this. A specification, is then the combination of it's context conditions; and the weights; utility function selectors; and goals; which are applicable under those context conditions.

Whether the specification is applicable depends on the context conditions being met. This may not be as simple as an equality relation. The aspect may not take a simple numeric type – it may describe abstract notions, or geographic spaces. Specifications are applied if *all* the context conditions are met according to their matching function by some context aspect in the overall context. The matching function may optionally define whether an unknown value matches true or false (false being default).

3.1 Combining Overlapping Specifications

There may well be specifications deriving from different aspects of context, referring to specifications giving different weights for the same type, different utility functions for the same attribute etc. Which should be chosen?

a. For type weights we choose:
   - Where one weight refers to a sub-type of another weight, the most specific type applicable to the element is applied.
   - Where the types are equal, a zero rating is honoured, otherwise the best is taken.

As type weight controls element inclusion and degradation order, the generous approach seems most valid – a preference is to be honoured. The only exception is a weight of zero, which specified “absolutely unwanted”. A weight of 1 might indicate *required* or a strong *preference*, depending on the selection algorithm employed. Our approach does not interfere with either approach. Given a moderate “default” value positive and negative preferences can both be described for more specific types.

b. For utility functions, all the applicable functions for an attribute are combined by product. Any zero utility result will clearly override all other values.

c. For goals, we take the lowest value for each resource, such as deadline. This will then reflect any limitations in time, screen space etc. which are being described.

We do not expect non-technical end-users to write these specifications. They could be provided by specialists, from user studies, and a user might tune parameters.

4 Conclusions

We have described a system of specification for adaptive applications using complex structured data, such as maps. Map data contains a wide range of semantic information, and is clearly useful in a range of mobile and ubiquitous computing scenarios. Our system has a flexibility and richness of specification which enables users with a wide and emerging range of devices, used in a diverse range of contexts, to achieve many different tasks, in a predictable and useful manner.

The division of parts of the specification according to *aspects of context* removes the need for describing specifications for every overall context which might be encountered. We have outlined a means of composing specifications to reflect a highly diverse range of contextual variation. This then enables *context mediated interaction* – the context does not trigger action, or associate itself with stored data, but is used to select the most appropriate data and behaviour. The division of the specifications enables them to be written, managed and applied in an effective way.

4.1 Acknowledgements

We gratefully acknowledge financial support from the EPSRC (Grant GR/M 26428 “Link-me” project).

References

1. M. Weiser, “Some Computer Science Issues in Ubiquitous Computing” *Communications of the ACM*, vol. 36, pp.75-85, 1993.
2. B. N. Schilit, N. Adams, and R. Want, “Context-Aware Computing Applications” *WMCSA*, 1994.
3. G.D. Abowd, E.D. Mynatt, “Charting Past, Present, and Future Research in Ubiquitous Computing” *ACM Trans. Computer-Human Interaction* vol.7, no.1, pp.29-58, March 2000
4. D. Chalmers and M. S. Sloman, “A Survey of Quality of Service in Mobile Computing Environments” *IEEE Communications Surveys*, *http://www.comsoc.org/pubs/surveys/*, Vol.2, No. 2. 1999.
5. D. Chalmers, M.S. Sloman, N. Dulay, “Map Adaptation for Users of Mobile Systems” *WWW-10*, Hong Kong, May 2001.
6. K. Holtman and A. Mutz, “Transparent Content Negotiation in HTTP” *IETF RFC 2295*, 1998.
7. W3C, “Composite Capability / Preference Profiles (CC/PP): A User Side Framework for Content Negotiation” *W3C note CCPP 19990727*, 27th July 1999.
8. A. Fox, S. D. Gribble, Y. Chawathe, and E. A. Brewer, “Adapting to Network and Client Variation Using Infrastructural Proxies: Lessons and Perspectives” *IEEE Personal Communications*, 1998.
9. Dublin Core Metadata Initiative *http://purl.org/dc*
10. W.-Y. Ma, I. Bedner, G. Chang, A. Kuchinsky, and H. J. Zhang, “A Framework for Adaptive Content Delivery in Heterogeneous Network Environments” *MMCN’00*, San Jose, California, USA, 2000.
11. R. Mohan, J. R. Smith, and C.-S. Li, “Adapting Multimedia Internet Content for Universal Access” *IEEE Trans. Multimedia*, vol. 1, pp.104-114, 1999.
12. A. Schmidt, K.A. Aidoo, A. Takaluoma, U. Tuomela, K. Van Laerhoven, W. Van de Velde, “Advanced Interaction in Context” *HUC’99*, pp.89-101, Karlsruhe, Germany, 1999.
13. A.K. Dey, G.D. Abowd, “Towards a Better Understanding of Context and Context-Awareness” *Workshop on The What, Who, Where, When, and How of Context-Awareness*, *CHI 2000*, The Hague, The Netherlands, April 2000.
14. P. J. Brown, J. D. Bovey, and X. Chen, “Context-aware applications: from the laboratory to the marketplace” *IEEE Personal Communications*, vol.4 no.5 pp.58-64, October 1997.