A Proposed Model for a Context Aware Distributed System

Duncan Bates[1], Prof. Nigel Linge[1], Dr. Martin Hope[1], Prof. Tim Ritchings[2]

[1] Centre for Networking and Telecommunications Research,
University of Salford, Salford, Manchester M5 4WT, UK

[2] Centre for Computer Science Research, University of Salford, Salford, UK

Abstract. Our paper draws on research work conducted in association with the UK electricity supply industry to design and deliver a context aware application for field engineers. This research identified clear business benefits and the need for intelligent information flow algorithms, which could match information delivery requirements to the capability of the delivery networks. In turn, our work has now resulted in a proposed generic model for the design and operation of a mobile context aware distributed system. The core objective of this research is the derivation of an algorithm to allow the efficient delivery of information to a user, based on the user's context. The pseudo-code for this algorithm is presented, together with an overview of the inputs to this algorithm, of which user context is one, and the issues surrounding them. Finally there is a look towards the future and possible further refinements that might be performed on the algorithm, namely the application of fuzzy logic. …

1 Introduction

The increasing use of mobile computer systems such as notebook personal computers, handheld PDA's and ubiquitous computing are by definition, presenting the user with a dynamically changing environment, where the context of that user and the context awareness of the mobile computer system are becoming increasingly important. For example: contextual information can be used in establishing ad-hoc networks, in routing protocols [1, 2], and to optimise network traffic. In addition, it can be used to find nearby resources such as printers, speakers, and video cameras and from an application perspective; it can make graphical interfaces more user friendly and flexible through the autonomy and discovery systems that contextual information can provide.

The work presented in this paper builds on previous work into the development of a context aware application for field engineers in the electricity supply industry [3]. This led to the development of a more generic model that is able to exploit a user's context in order to prioritise information flow [4].

Bates D., Linge N., Hope M. and Ritchings T. (2004).
A Proposed Model for a Context Aware Distributed System.
In *Proceedings of the 1st International Workshop on Ubiquitous Computing*, pages 180-194
DOI: 10.5220/0002679301800194
Copyright © SciTePress

1.1 Context Awareness

To fully appreciate what constitutes context awareness and what context actually is, Dey [5] provides a comprehensive review of the most prominent work that has previously attempted to define context and contextual awareness. These definitions are probably the most concise and well defined to date and are as follows:

Context: *'Context is any information that can be used to characterise the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves'*.

Context aware: *'A system is context aware if it uses context to provide relevant information and / or services to the user, where relevancy depends on the user's task'* [5].

Dey and Abowd also propose a hierarchy of contextual types and outline a method of categorising context aware applications. Identity, Location, Activity, and Time are considered as the four primary context types [6]. These primary context types are then used to index and define secondary context types. For example: given a persons' current location, a secondary context type could be: where they have been and where they are going. This hierarchy obviously has applications to defining a class structure of context types that could be used in the development of context aware applications, or indeed an xml based mark-up language, as has recently been attempted [7].

The rest of this paper presents the proposed model and the justification for it within the framework of the current literature (section 2). The algorithm at the heart of the research is presented in full in section 3, together with a brief introduction to fuzzy logic. Finally the paper is drawn to a conclusion in section 4.

2 A Generalized Model for a Mobile Context Aware System

With the significant growth in mobile communications, the ability to determine location is leading to the development of a new generation of multimedia location based services. However, as other authors have previously pointed out, context is more than just location [8]. Today's wireless networks range from low bandwidth GSM through 3G to IEEE 802.11 WiFi. In the main, network bandwidth forces a limitation on the deployment of wireless based interactive multimedia applications. Therefore there will always be a mismatch between the volume of content and the available bandwidth to deliver it to the users at a location convenient to them. This will lead to a need for new information processing systems, which are able to understand completely the user's context and appreciate the capacity of their network access technology. In this way information is prioritised to ensure that the most relevant and valuable information is delivered in the most time efficient manner.

A generic model for relating distributed information sources to a user's context is proposed. The tasks performed by a user are defined by a workflow process model. Each user is defined in terms of their user context, which must also take into account network availability. Given these three inputs it is then possible to create an optimised information flow algorithm which can ensure a quality of delivery which de-

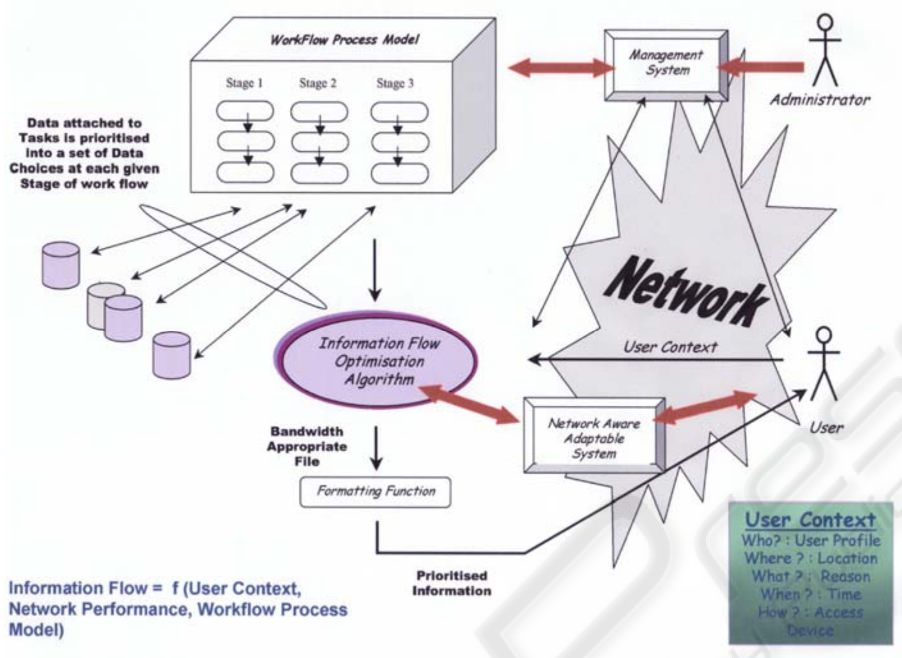

Fig. 1. A generalised model for a mobile context aware system

termines the most relevant piece of information and delivers this to the user in a format that makes best use of the quality of their network connection.

2.1 User Context

The proposed model is illustrated in figure 1. As applicable to the model, user context is here shown as made up of the elements, user profile, location, reason, time & access device. The first four of those elements can be matched against Dey and Abowd's generalized definition. These are characterized simplistically in our model as "who, where, what & when. A further element has been added to the definition, namely the access device, and this can be characterized simplistically as "how".

User profile represents all that is pertinent about the user, i.e. the user's id, name, address, age etc together with the authority that the user would have within a given company and the expertise of that user. Location is self-evident and would likely be drawn from network data, which is related to known geographic locations. Reason (or activity) is that which the user is doing at a given point in time or most likely just the job that has been assigned to that user. Time has some bearing as a piece of user context because in applying it to a workflow process model, certain activities may vary according to the time of day or night. Finally the access device is an important piece of user context because the format & size of content that we might want to deliver back to the user would be dependent on the type of device with which the user is receiving that content.

2.2 The Workflow Process Model

The Workflow Management Coalition, probably the primary standard body in this area, in their Workflow Reference Model [9] describe workflow management as follows:

"*Its primary characteristic is the automation of processes involving combinations of human and machine-based activities, particularly those involving interaction with IT applications and tools*."

Research in this area draws upon and feeds into this reference model, and the model tends to be used as a template for ongoing work. Gillmann et al [10] describe Workflow Management Systems (WFMSs) as *'geared for the orchestration of business processes across multiple organizations'* and comment that they are *'complex distributed systems'*.

Kappel et al [11] suggest a framework for WFMSs based on objects, rules and roles. The objects in question refer to an object-oriented model, and in addition there is a role model and a rule model. The role model concept is common throughout much of the literature in order to provide a separation between the activities that agents perform and the agents themselves, where agents can be people or hardware/software systems.

Muehlen [12] discusses resource modelling in workflow applications. Resource Models are models of people & systems and as such closely matches the concepts described by Kappel et al. Similarly assignment of activities or tasks is by role, and Muehlen postulates that *'the main purpose of the role model is the separation of workflow and resource model'*. There is however, intimation towards the need for error or exception handling. The author quotes directly from another author, Sachs [13] who states that *'the efficiency of work is less dependent on the structure of the workflow than on the exception handling capabilities of the resources involved'*.

Chiu, Li & Karlapalem [14] address what they feel is a fundamental deficiency with this and other work in the research area. They suggest that a simple role model does not go far enough towards matching the reality within the workplace. They not only talk of roles, but capabilities, thereby drawing the distinction that although an agent may perform a certain role, their capabilities for a given activity may vary.

There is clearly considerable related work in the workflow community. However, the fundamental contribution that this paper makes is the application of user context to the workflow management system. As noted by Gillmann et al, WFMSs are complex systems. Our proposed model is an over-arching one across several disparate areas of research. Therefore it was felt that a faster approach to describing workflow processes could be made using UML [15, 16] as a modelling language and designing a simplified workflow model to fit the needs of the system. It is simplified in that it does not consider the worklists of agents, agent assignment by the system and exceptions are generally propagated up to and handled within the presentation layer. It could also be termed a role-based model.

Note that the term that has been adopted for our simplified workflow model and henceforth in this paper is the workflow process model. The workflow process model can be described in terms of the discrete stages able to be performed by a given worker's role. Furthermore those stages are then broken down into the tasks required to complete that stage of the work. Attached to some of these tasks is data in the form of a set of data choices, which roughly represents the separation of the same content into a number of different presentations. The data may be distributed in that it may, for example, reside in different databases or data sources on different machines.

As part of the research, we are examining a small number of case studies to describe their workflow processes. The electricity supply industry field engineer workflow processes were documented within the report that went with that project, and these act as one of the case studies. In addition we have other companies and organisations on board that primarily employ mobile working practices, and the accumulation of further case study material is currently being investigated. Eventually when enough case study material has been gathered, this will allow the research to work towards a truly generic solution by testing the system against these different inputs.

2.3 Network Availability

If data is attached to a task, at the point of delivery, the information flow optimisation algorithm needs to determine the state of the network. This has an enormous bearing on the final format in which the data is delivered, in that it allows the algorithm to determine which data choice to send. If the network is very congested it's likely that a large file such as a video file could not be delivered within a time period that would be acceptable to the user. The video file might possibly represent the top data choice, therefore a lesser choice must be chosen.

The adaptable system needs to look at a number of network parameters, such as throughput and latency to determine current data bandwidth availability and potential information delivery times. It should be noted however that the adaptable system is just taking a snapshot of the network. An instant later the network traffic could have changed and certainly this might be the case by the time the chosen data choice has been formatted and delivered back to the user. To circumvent this, the adaptable system must take measurements over a greater period of time and average the results or recognise a trend.

2.4 Information Flow

At the heart of the model is the information flow optimisation algorithm, which is now the main focus of our research. This algorithm takes the three inputs: user context, the workflow process model and network availability and in its simplest execution determines whether a given (workflow) task should be returned to the user. In its more complex form, if data is attached to the task, it determines the most bandwidth appropriate of a set of data choices to return to the user.

The set of data choices, attached to any of the given tasks, range in format from the most bandwidth hungry choice down to the simplest and smallest. Typically, the first choice might be a video file, which is of the order of megabytes, down to the last choice that might simply be a text file, of the order of a few kilobytes. As far as is applicable, the files that are delivered are initially in the form of an xml file together with associated xsl files. In this way, the data content is kept separate from its presentation, and the same data can be presented in a number of different formats. For example the format required if the user is logged on to a web page via a laptop (html) would be different from that required if the user had a wap enabled GSM mobile (wml 1.x) as his/her access device. The formatting function applies the xsl stylesheet to the xml file to perform the xsl transformation (xslt) and the result is then displayed in the appropriate format, either html, wml 1.x for GSM mobiles, or in the near future, xhtml for 2.5G and 3G mobiles [17].

3 Information Flow Optimisation Algorithm

Although not presented here, the model above has been fleshed out into a complete J2EE design [18], divided into two parts that are intended to be implemented and deployed on two separate J2EE compliant servers. As such this represents the delivery system for the information flow optimisation algorithm. It is the derivation of this algorithm that is the primary goal of our research, in order to achieve the efficient delivery of information back to the mobile user.

As previously stated the primary inputs to the algorithm are user context, the workflow process model and network availability. The elements of user context that are used most prominently and directly within the algorithm itself are user authority, and user expertise, which are tested against equivalent values on the individual tasks and attached data. However, other elements of user context are not redundant, for example, the user job is considered to be an instance of and is matched against a workflow process model stage. User location determines proximity to nearest job (location) and is needed to return tasks back to the user.

Figure 2 illustrates the structure of the workflow process model side of a system that could be implemented to satisfy the proposed model. This side of the system is made up of the classes that reside on the workflow process model server. Primarily, these classes conform to the J2EE specification, in that they are grouped into packages of 3 that together make up an enterprise java bean (ejb). An ejb is made up of its

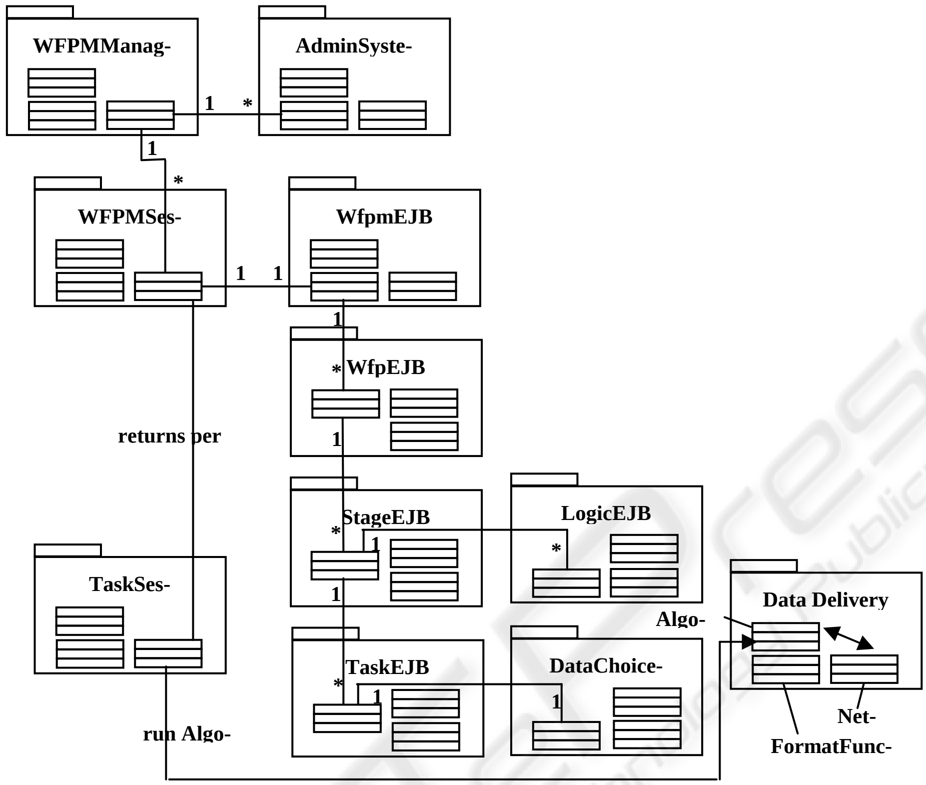

Fig. 2. Generic Structure of the objects on the Workflow Process Model Server

bean implementation, its remote or local interface & its home interface. The detail of these classes is not presented here, but briefly enterprise java beans are divided into session beans (which come in two forms, stateless & stateful) and entity beans. To quote the J2EE Developer's Guide [18] "*Entity Beans represent business objects in a persistent storage mechanism such as a database".*

The entity beans in the generic workflow process model above are WfpmEJB, WfpEJB, StageEJB, TaskEJB, LogicEJB & DataChoicesEJB and these are a reflection of the underlying database tables. There is a 1:Many relationship between the ejb's WfpmEJB & WfpEJB, WfpEJB & StageEJB, StageEJB & TaskEJB, which reflects the underlying database tables. So in other words for each workflow process model (wfpm) there are many workflow processes (wfp), and for each workflow process there are many stages etc. A stage represents a point at which the user, assigned to a given job, can be first directed to upon logging onto the system. This is the equivalent of saying that the user is now performing a set of tasks that carry out a stage of his/her given workflow process and wants to be given the first task in that sequence. This in effect is the reason that the user is performing the task, and is represented simplistically in the user context as 'what' the user is doing.

The structure presented represents a generic structure that should allow all workflow process models to be fitted into it. However there is a great deal of variation within workflow process models and it would seem that this structure is too general. In order to overcome this, specific workflow process models plug into the generic structure at the level of the set of tasks. Each ‘generic task’ has a redirect URL as one of its attributes / fields thus allowing the generic set of tasks to redirect to a real set of tasks. The real set might require extra database tables and thus extra classes or ejb’s in order to process the information from them.

Returning to the generic structure, the LogicEJB represents a set of logic records that tell the tasks how to behave, i.e. the order & sequence of tasks that a user must have delivered, according to the decisions that he/she makes. This logical sequence, basically follows the permitted paths, splits & joins that are allowed in a UML activity diagram, e.g. decision points & parallel processing splits & joins. Note that within the workflow community, this would generally be referred to as activity ordering. In our work the logic records are just a short hand way of representing those paths. A system implementation would need to have a class that could handle the processing of those records back into a logical sequence of instructions for the delivery of a set of tasks to a user.

Each task in a stage can have attached data, which as identified earlier, is made up of a set of data choices ranging from the highest bandwidth option down to the simplest choice, which is normally a plain text file. Not all tasks have attached data, and this would be indicated by a flag as one of the attributes on a task ejb (or field on the task record). A DataChoicesEJB holds the state of one database record, which stores the URL’s of the files that would need to be fetched when required.

Finally the Data Delivery package is made up of two ordinary Java classes and a Java Servlet. Of prime concern is the Algorithm class which performs the routines referred to in the earlier section. As noted, the main inputs to the algorithm are user authority & user expertise, task authority & task expertise, user location, the access device & a set of data choices if data is attached to the task to be delivered. The algorithm then determines whether the task should be delivered, and if data is attached which data choice can be delivered. This is determined in turn by calling the NetAwareAPI, which calculates the network availability. If the data choice is made up of an xml file & xsl file, the xslt transformation is performed by forwarding the request to the Java Servlet, FormatFunction.

In an earlier paper we posed a set of key questions that were asked of the design and the algorithm in order to ensure consistency and its ability to withstand all scenarios [19]. Among the ideas and conclusions that came out of this process were

The possibility of a data choice matrix, i.e. of 2 dimensions, rather than a simple linear set of data choices. This is expressed across the user expertise axis as well as down the data choice (required bandwidth) axis. As a simple example, it might be possible to deliver a video file to the expert user, but he/she might prefer an advanced succinct text file.

The need for a set of advanced tasks at higher levels of user expertise in order to compensate for the non-delivery of 3 or 4 basic tasks and in particular to maintain a coherent ‘story of events’ delivered back to the user.

The information flow optimization algorithm can be synthesized down into a fairly simplistic form. The issues mentioned above, along with the authority, task expertise and data expertise tests are presented in figure 3 as pseudo-code. An explanation of terms precedes the pseudo-code.

3.1 Explanation of Terms

(a) Data rates vary according to access device & network. A typical range of data rates chosen for the algorithm are down from a pc on a 10Mbps Ethernet to a GSM mobile with typical rates of 9.6 kbps.

(b) Similarly the data choices represent a typical range of files that might be delivered to the user, ranging from a video file, of the order of 20Mb down to a plain text file, of the order of 10kb.

(c) The expertise test includes a comparison of user expertise against a maximum task expertise and a minimum task expertise. The reason for this is that as a user gains in expertise he/she requires only those tasks that are applicable to his/her expertise. However, as a user’s expertise gets more advanced, it’s necessary to supplement the workflow process model with advanced tasks, otherwise workflow meaning is lost. This is also consistent with working practices, in that sometimes expert users tend to skip tasks in a workflow process because they feel that they know the job, but in doing so follow incorrect procedures through over-confidence. In practice, advanced tasks can be supplemented in place of 3-4 more elementary tasks.

(d) The task is simply a textual instruction, but would need to be formatted appropriately for the given access device (e.g. html for a web browser, wml 1.x for a GSM mobile).

(e) This test corresponds to the data choices matrix. Having previously determined which horizontal axis the data choice lies along (i.e. data choice 1-4), the algorithm must choose the format appropriate to the user’s expertise along that axis.

(f) Although a typical set of data choices might be as previously stated, video file (1), sound file (2), text file with graphics (3), plain text file (4), in practice all these formats will probably not all be available against a given task. Moreover, they might be available for one expertise of user, e.g. the novice user, but not another. Therefore the highest available choice should be delivered, which is determined as explained above.

(g) If there's no data attached to a task, the only processing that the algorithm has to make is whether the authority & expertise tests have been passed. If so, the task is delivered in the appropriate format for the access device. Network availability is not a consideration in this case because a task is typically a textual instruction to the user.

<u>Data Choice Calculation</u>

```
while (Delivery Flag is false) {
        // Data Choice 1
        Download Time = (DATA_CHOICE_1_SIZE) /  (Data Rate x Network
        Availability)
        if (Download Time <= Cut Off Time) {
                Delivery Flag = true
                Data Choice = DATA_CHOICE_1
                break
        }

        // Data Choice 2
        Download Time = (DATA_CHOICE_2_SIZE) / (Data Rate x Network
        Availability)
        if (Download Time <= Cut Off Time) {
                Delivery Flag = true
                Data Choice = DATA_CHOICE_2
                break
        }
        …. Etc
}
```

```
Access Devices [1 = PC, 2 = Laptop, 3 = 3G (Indoors), 4 = 3G (Outdoors), 5 =
GPRS, 6 = GSM] - (a) Data Choices [1 = Video, 2 = Sound, 3 = Text with
Graphics, 4 = Plain Text] – (b)

// Authority Test
if (User Authority >= Task Authority) {
        // [TRUE] Continue

        // Task Expertise Test - (c)
        if ((Task Expertise Min <= User Expertise ) && (User Expertise <=
        Task Expertise Max)) {
                // [TRUE] Continue

                if (Data Attached) {
                        // [TRUE] Continue
                        Deliver the Task in the format appropriate for the De-
    vice – (d)
                        Consider the Network Availability (*)
                        Calculate the Data Choice (1-4)  (*)

                        // Data Expertise Test – (e)
                        For given data choice (e.g. 1) …
                           if ((Data Expertise Min <= User Expertise ) &&
                        (User Expertise <= Data Expertise Max)) {
                                (Highest) Data Choice = Appropriate format
                                along User Expertise axis,
                                e.g. 1 (Novice), 1 (Skilled) or 1(Expert)
                        }  // Data Expertise Test

                        // Availabilty Test – (f)
                        if (Highest Data Choice Available) {
                                Deliver highest Data Choice
                        }
                        else {
                           Firstly go back across User Expertise axis & de-
                liver the less expert data choice format, e.g.
                1(Novice) instead of 1 (Skilled) if available
                        Then descend the data choices, considering firstly the
                format appropriate to the user's expertise,
                then the lesser format, e.g. if the user is
                skilled… 2 (Skilled) then 2 (Novice) then 3 (Skilled) then
                3(Novice) etc etc until a data choice is available,
                then Deliver it !
                        }  // Availability Test
                }
```

3.2 Fuzzy Logic

Fuzzy logic was introduced to the world by Lofti A. Zadeh in 1965 in his seminal paper *Fuzzy Sets* [20], and is best summed up by Zadeh's law of incompatibility which states that

"As complexity rises precise statements lose meaning and meaningful statements lose precision".

Fuzzy Logic can be thought of as a superset of conventional boolean logic that has been extended to handle the concept of partial truth. So instead of traditional boolean logical variables, such as 0 and 1 or true and false, there are linguistic variables. Linguistic variables embody the concept of imprecision or fuzziness, exemplified by such terms as 'hot', 'warm' and 'cold', 'near' and 'far' or 'tall' and 'short'. These terms mimic the way humans interpret linguistic values, and also imply imprecision and would depend on the context to which they are applied [21]. Instead of hard precise definitions, there are fuzzy sets, e.g. the interval [0,1], and an input variable is mapped to a linguistic variable by the so-called process of fuzzification. A so-called fuzzy set A, couples each element $x \in X$ with a membership function $\mu_A(x)$ that indicates the membership degree of the element x for the set A [22]

.

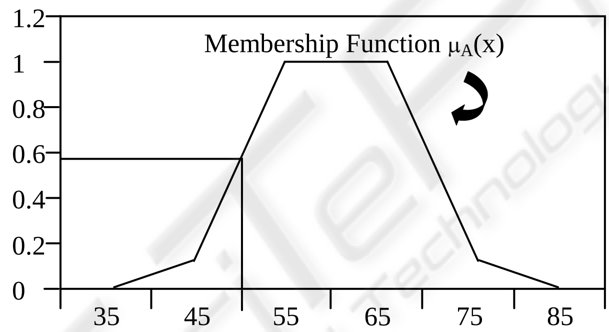

Fig. 3. Membership Function for a temperature x of a Thermometer X

As an example, figure 3 is extracted from the same source [22] as above, as is the explanation below.

"If X is a continuous set then A is represented by its membership function. Fig.3 shows the membership function for the temperature x of a thermometer X. For x = 60 °F, the temperature is in A.

For x = 50 °F, the temperature is only partially in A. It belongs to A with a degree of 58%."

In other words X represents the set of all possible temperatures measured by a thermometer and A is a subset of X. If A was a crisp subset it would either contain or not contain a particular element, $x \in X$, and the membership function would have two values, 0 and 1 across the set to indicate this. However if A is a fuzzy set as shown in figure 3, each element, $x \in X$, is coupled with its membership function, $\mu_A(x)$ which indicates the degree of membership of that element for the set A. The example x = 60 ^{o}F is fully in A because the membership function is 1 at this point, in contrast to x = 50 ^{o}F, where it is 0.58. Clearly there is much more to fuzzy logic than this simple example, but hopefully this has served as a brief introduction to the subject.

The design of the proposed model discussed in this paper is largely borne of an object-oriented methodology, and indeed as mentioned earlier, it is intended to be implemented as a J2EE design. J2EE, although particularly suitable as architecture for developing, deploying, and executing applications in a distributed environment [18], still has Java code at its heart, which is an object-oriented language. Essential artefact types in object-oriented languages are entities, classes, attributes, operations, and aggregation and inheritance relations. Contained somewhere within those operations are likely to be conditional statements of the form

If <test> then <consequence>

which may or may not act on the attributes within instances of the classes [23].

In our hard and fast logical tests in the pseudo-code in figure 2, the variables for task expertise, user expertise etc take pre-defined quantized values from 1 to 5 and the tests are simple comparisons of those values. For future work, we would like to 'fuzzify' the logic in the algorithm by substituting those values with, for example, linguistic variables such as 'novice', 'skilled' and 'expert'. These particular examples, of course, apply to user expertise. In this way, we will be able to determine whether the intuitive design and consequent derivation of the algorithm arrived at by these human authors would differ in any way from that arrived at by fuzzy logic.

4 Conclusion

A proposed model for a distributed context aware system has been presented based on experience gained from the development of a context aware application for field engineers in the electricity supply industry. The fundamental elements of this model were examined in detail, including the three major inputs to an information flow optimisation algorithm, namely user context, a workflow process model and network availability.

User context is made up of a user profile, user location, the time, user activity (reason or most likely the user job) & his/her access device. This is constructed and delivered to the workflow process model. The workflow process model is made up of stages & tasks. To the stage is attached the logic that governs the tasks, and to the task a set of data choices can be attached.

Elements of the three major inputs are presented within the pseudo-code for the information flow optimisation algorithm. This is basically made up of a series of if-then scenarios, which act as tests in order to determine the appropriate output to deliver back to the user. An example of such a test is that of the user expertise against task expertise. This particular test is complicated slightly by the need for the substitution of a set of advanced tasks for every 3 or 4 basic tasks, in order to maintain a coherent chain of tasks back to the user. Lookups against network availability are also performed and the most appropriate data choice deliverable within a given cut-off time is determined.

Although it may not be necessary to build a full working system, it would be hoped that at the least a prototype of the workflow process model server objects could be constructed, thereby allowing the nature of the algorithm to be tested. The case studies would provide 'real world' input and variation to examine the generic structure of the model on this side of the system. However, in order that we don't simply build algorithms that provide an adequate solution for the model we have constructed, an alternate approach is considered.

Since the tests within the algorithm, on the whole conform to simple rule based if-then scenarios, there was felt to be some scope for the exploration of a fuzzification of those rules. In other words, if fuzzy logic could be applied to the inputs to the algorithm, then maybe a different outcome would appear from that chosen by the authors. This now forms the basis of future work that is in the planning stage for the algorithm. Comparisons will then be able to be made between the intuitive, 'common sense' design for the algorithm and the machine based conclusions, albeit based on inputs that mimic human linguistic concepts.

One of the major conclusions to draw from this analysis is that it is clear that workflow process models have to be highly documented, as regards the tasks, attached data and the logic that governs the sequences of tasks that are delivered to the user. An administrator has to pre-load all of this, including such additional information as task expertise, data expertise and user context, specifically relating to authority and expertise levels. Users must also be assigned jobs by the system, which are instances of a previously documented workflow process model. A full system therefore relies on all this information in order to match a user job against a set of tasks. However, as an exercise in marrying together the disparate elements of user context and 'real world' workflow process models, our proposed model is a powerful demonstration that we can move towards the creation of a system that is truly context aware.

References

1. Veeraraghavan, M., Pancha, P., Eng, K.Y., 1997. Application-aware routing protocol. In *ISCC 97, Second IEEE Symposium on Computers and Communications.* Alexandria, Egypt, July 1-3, 1997. pp 442–448.
2. Basagni, S., Chlamtac, I., Syrotiuk, V.R., 2000. Location aware one-to-many communication in mobile multi-hop wireless networks. In *VTC 2000, IEEE 51st Conference on Vehicular Technology*, Spring Tokyo, Vol 1, 2000, pp 288–292.
3. Hope, M., Linge, N., 2000. Improving co-operative working in the Utility Industry through mobile context aware geographic information systems. In *ACMGIS 2000, 8th International Symposium of Geographic Information Systems.* Washington DC, 10th–11th November 2000. pp 135–140.
4. Bates, D., Linge, N. Hope, M., 2002. Context Aware Information Services for Mobile Workers. In *PGNet 2002, 3rd Annual Postgraduate Symposium on The Convergence of Telecommunications, Networking & Broadcasting, 17-18th June 2002, Liverpool John Moores University*, ISBN 1 902560 086, pp 390-395.

5. Dey A.K, 2001. *Understanding and Using Context.* Personal and Ubiquitous Computing, Vol 5, No 1, pp 4-7.
6. Dey A.K., Abowd G.D., 2000. Towards a better understanding of context and context-awareness. In *CHI 2000, Conference on Human Factors in Computing Systems.* The Hague, The Netherlands, April 3, 2000.
7. Strang T. & Linnhoff-Popien C., 2003. Service Interoperability on Context Level in Ubiquitous Computing Environments. In *Proceedings of International Conference on Advances in Infrastructure for Electronic Business, Education, Science, Medicine, and Mobile Technologies on the Internet.* L'Aquila, Italy, January 2003.
8. Schmidt A., Beigl M. & Gellersen H., 1999. *There is More to Context than Location*, Computer & Graphics, Vol 23. (1999) pp 893-901.
9. Hollingsworth D., 1995. *The Workflow Reference Model.* [online]. TC00-1003 Issue 1.1, Workflow Management Coalition, Jan 1995. Available from: http://www.wfmc.org/standards/docs/tc003v11.pdf [Accessed 4th Dec 2003].
10. Gillmann M., Weikum G. & Wonner W., 2002. Workflow Management with Service Quality Guarantees. In *Proceedings of the ACM SIGMOD Conference on Management of Data*. Madison, Wisconsin.
11. Kappel G., Rausch-Schott S. & Retschitzegger W., 2000. *A Framework for Workflow Management Systems based on Objects, Rules and Roles.* ACM Computing Surveys, 32 (1), 2000.
12. zur Muehlen, M., 1999. Resource Modeling in Workflow Applications. In *WFM99, Workflow Management Conference*, Muenster, Germany, November 9th 1999, pp 137-153.
13. Sachs, P., 1995. *Transforming Work: Collaboration, Learning and Design.* Communication of the ACM, 38 (9), pp 36-44.
14. Chiu D.K.W., Li Q. & Karlapalem K., 1999. *A Meta Modeling Approach to Workflow Management Systems supporting Exception Handling.* Information Systems, 24(2), 1999, pp 159-184.
15. Bennett S., Skelton J. & Lunn K., 2001. “Schaum’s Outlines – UML”, McGraw-Hill, New York.
16. Holt J., 2001. “UML for systems engineering – watching the wheels”, IEE Professional Applications of Computing Series 2, London.
17. Wireless Application Protocol Forum Ltd., 2002. *Wireless Application Protocol, WAP 2.0, Technical White Paper.* [online]. WAP Forum. Available from: http://www.wapforum.org/what/WAPWhite_Paper1.pdf [Accessed 4th Dec 2003].
18. Sun Microsystems Inc., 2000. *The Java™ 2 Enterprise Edition Developer’s Guide, Version 1.2.1.* [online]. Sun Microsystems Inc. Available from: http://java.sun.com/j2ee/sdk_1.2.1/devguide1_2_1.pdf [Accessed 4th Dec 2003].
19. Bates, D., Linge, N. Ritchings, T., Chrisp T., 2003. Designing a Context-Aware Distributed System – Asking the Key Questions. In *CIIT 2003, 2nd IASTED International Conference on Communications, Internet and Information Technology*. Scottsdale, Arizona, Nov 17-19th 2003. pp 361-371.
20. Zadeh L.A., 1965. *Fuzzy Sets*, Information and Control, Vol. 8, No. 3, pp 338-353.
21. Zadeh L.A., 1994. *Fuzzy Logic, Neural Networks and Soft Computing*, Communications of the ACM, Vol 37., Issue 3 (March 1994), pp 77-84.
22. Benedicenti, L., Succi, G. Vernazza, T., Valerio A., 1998. Object Oriented Process Modelling with Fuzzy Logic. In *Proceedings of the 1998 ACM Symposium on Applied Computing.* Atlanta, Georgia, USA. ISBN:0-89791-969-6, pp 267-271.
23. Marcelloni F., Aksit M., 2000. *Improving Object-Oriented Methods by using Fuzzy Logic,* ACM SIGAPP Applied Computing Review, Vol. 8, Issue 2 (Fall 2000), pp 14-23.