Ad-Me: A Context-Sensitive Advertising System

N. Hristova
University College Dublin
Belfield
Dublin 4, Ireland
Nataliya.Hristova@ucd.ie

G.M.P. O'Hare
University College Dublin
Belfield
Dublin 4, Ireland
Gregory.Ohare@ucd.ie

Abstract: *The mobile commerce sector and in particular the context sensitive advertising will represent a high yield revenue stream. This paper introduces the Ad-me (**Ad**vertising for the **M**obile **E**-commerce user) system. The Ad-me is a mobile tourist guide that proactively delivers advertisements to users based upon perceived individual user needs together with their location. A Multi-Agent Systems (MAS) design philosophy is adopted. In order to achieve maximum content diffusion a range of presentation formats are accommodated including HTML, WML, HDML and iMode.*

1. Introduction

Mobile commerce is set to witness phenomenal growth. CNET News predicts that by 2005 there will be 1 Billion mobile devices world-wide, more than the number of cars and PCs combined. In Europe and Japan, the early adopter market, $400 Million per annum is currently generated in revenue from m-commerce [Irish Computer April 2001]. According to an Ovum report in 2001, M-commerce information services will generate $4.7 Billion by 2004 only to be surpassed by advertising, which will generate $8 Billion by 2005. Furthermore, mobile location services are set to increase from 2 Million connections in 2001 to 560 Million in 2006 with maximum revenue streams derivable from services that are personalised and easy to use. Tourism and advertising constitute highly attractive m-commerce market segments.

This paper introduces the Ad-me (ADvertising for the Mobile E-commerce user) system, a context sensitive advertising service for the mobile user. The Ad-me system sits on top of a

mobile tourist guide where the motivation and added value service offered to the user is that of context sensitive tourist services accommodated upon a Personal Digital Assistant (PDA) or cellular phone. The tourist content thus provides a carrier mechanism for the true objective of the system, that of targeted advertising, delivering advertisements to the user *when* they need them, *where* they need and in a form sensitive to their technological context (*how* they need them). Differential advertising tariffs and priorities result in varying levels of penetration to the user community. The Ad-me system strives for maximum diffusion and can thus be hosted on any wireless device supporting either Wireless Mark-up Language (WML), Handheld Device Mark-up Language (HDML), HyperText Mark-up Language (HTML) or compact HTML. It operates in an outdoor environment and obtains user location based on a Global Positioning System (GPS) receiver.

The remainder of the paper is structured as follows: Section 2 situates our work among other related research. Section 3 describes our design philosophy and Section 4 considers some computational constraints unique to PDAs and mobile phones. Our choice of implementation technology is justified in Section 5. The Ad-me functionality is outlined in Section 6 while the architecture is described in the penultimate section. Section 8 presents our conclusions.

2. Related Work

Context-sensitive electronic tour guides are not new. There are a number of research projects related to this area. Pioneered context-sensitive systems are ParcTab [2], Olivetti's ActiveBadge [23] and Cyberguide project [1]. The GUIDE project [9] developed a context-aware tourist guide providing visitors of Lancaster City with a structured tour based on a set of attractions they may wish to visit. A handheld tour-guide that dynamically delivers multi-media presentations based on the user's location and profiles has been developed by the HIPS (Hyper Interaction within Physical Space) project [19].

More recently some researchers have begun to deploy Multi-Agent techniques. Several such systems exist including Impulse, MIA and ComMotion. The Impulse project [24] provides the user with personalised location-based information with the assistance of an agent. A User Agent residing on the hand-held device assembles a user profile and builds queries for the Wherehoo server and Provider Agents. User Agents select and supply user with relevant URLs. MIA [4] also constitutes a multi-agent based location-aware information system for mobile users. MIA's emphasis is one of retrieving content from the web and delivering this to the user. The retrieved information is relevant to the city where the user is currently located. It merely uses user location as the key. ComMotion [18] represents another location-aware computing environment. A location-learning agent observes the locations frequented by the user via a GPS receiver. It uses both speech and graphical user interfaces and provides map display, showing the user's current position together with the local neighbourhoods.

In contrast to mobile PDA hosted tour guides, context-sensitive advertising is at an embryonic stage. Two companies Streetbeam and Adlive both have just developed billboard advertising with PDA interactivity. Thus pointing the PDA will receive additional advertising content. At present, virtually no mobile context-sensitive advertising system exists. However companies like Doubleclick, AvantGo and Lot21 have or will deploy limited trials. Toward the end of 2000 a

new advertising agency called D2 Communications was capitalised in Japan to a value of 490 Million Yen which will specifically target advertising for iMode.

3. Agent Oriented Design

The fundamental objectives of the Ad-me system are:

- Provide a backdrop of tourist related information;
- Provide a context-sensitive advertising capability;
- Provide a simple and effective advert posting capability.

In achieving these objectives we adopted an agent-oriented approach [24] to the design of the system. Much research work has been commissioned on Multi-Agent Systems (MAS) and Distributed Artificial Intelligence (DAI). In the delivery of computationally tractable models of deliberative reasoning, one approach that has gained wide acceptance is to represent the properties of an agent using mental attitudes such as *belief*, *desire*, and *intention*. Multi-agent architectures that are based on these concepts are referred to as *BDI-architectures* (Belief-Desire-Intention) [20]. Proponents of the BDI approach argue that the understanding of the dynamics of these mental attitudes and their intimate interdependencies is crucial in achieving *rational* agent behaviour.

Currently, Multi-Agent Systems are viewed as an appropriate technology for the delivery of services to mobile and wearable devices. Commensurate with this is the need for lightweight agents with an associated small *software footprint*. Using a mixture of heavyweight and lightweight agents, where the later is hosted on the client devise improves the response time. Numerous systems have sought to deliver support for mobile lightweight agents including Tromso And Cornel Moving Agents (TACOMA) [15], [16] and Grasshopper [6], both however merely offer a weak notion of agenthood. The BDIM toolkit [7] however, offers support for Mobile BDI agents. Within the context of this work we will use Agent Factory, a system developed at University College Dublin (UCD), which also supports mobile BDI agents. Section 7 will provide a more detailed description of Agent Factory.

4. Mobile Device Constraints

There are four specific mobile device issues we must overcome namely interoperability, computational constraints, mapping technology and user localisation. We will discuss each of these briefly. The necessary interoperability demands a separation of concerns between presentation and content, supporting the full range of mobile device browsers: HTML, WML, HDML, Avantgo and iMode. iMode uses an extended subset of HTML called cHTML (compact HTML).

The computational constraints of mobile devices represent significant impediments. Furthermore memory and screen size limitations differ with each device, supporting different image formats (e.g. GIF, WBMP for WAP). Because of memory and bandwidth limitations, sophisticated rich value services cannot reside entirely on the client side. That is why we are going to host on the client side lightweight agent that is with minimal code and minimum of functionality. Various lightweight agents can act as flexible bridges between the complex agents. Using mixture of

complex and lightweight agents the response time and the availability can be improved [7], [13]. Lightweight agents can be written in Java. It is foreseen that soon the majority of mobile phones will be equipped with Java Virtual Machine (JVM). There are currently eighteen mobile phones with Java, while ten are having JVM for PDA.

The judicious choice of mapping technology is inextricably related to graphic content quality. Raster formats tend to be limited to a single, often low resolution and consume large amounts of bandwidth on the Web. Popular high-quality vector based graphic formats are Flash and Scalable Vector Graphics (SVG). Considering the limited *real estate* on a mobile, scaling and zooming map images is of paramount importance. Unfortunately there is, at present, no support for vector image formats on mobile phones.

The most suitable outdoor user localisation technique is the Global Positioning System (GPS) [5], [11]. The original Selective Availability (SA) degradation has recently been relaxed allowing an accuracy of 10 to 20 metres with normal GPS. The European Union GALILEO project aims to provide better precision, guaranteeing reliability and coverage levels not present in existing systems. The reasons for developing Galileo are both strategic and economic since it has been estimated [12] that the global market for satellite navigation systems and services will be worth $40 billion between now and 2005. Galileo will start operations in 2005 and achieve full operational status in 2008.

Accuracy of 1 to 5 meters can be achieved by using a differential correction comparing data received form a second GPS receiver of known location. Cellular base station triangulation represents an alternate approach. Such example is CURSOR, which is based on the BTCellnet E-OTC system. New systems use some combination of satellite and cellular techniques. The Enhanced Global Positioning System (EGPS) [SNAP] is one such system. Almost all leading network providers are developing their own form of location sensing technologies. Ericsson have developed the Mobile Positioning System (MPS) and indeed the European Telecommunications Standards Institute (ESTI) and the American National Standards Institute (ANSI) have decided to adopt a standard based upon this.

5. Implementation Technology

Technologies for dynamic content delivery can be classified into server-side and client-side. Client-side technologies place a heavy processing load on the client-side device. In server-side technologies, the browser is only given the resulting HTML/WML page generated on the server. This minimises network traffic between the browser and the server and makes server-side technologies more suitable for *thin* clients.

Four of the leading server-side scripting products providing dynamic content are Hypertext Pre-processor (PHP) [8], Java Server Pages (JSP), Mircosoft's Active Server Pages (ASP) and Cold Fusion (CF) [22]. Both ASP and CF are proprietary products. The former necessitates ASP developers to use a Microsoft web server such as IIS. Other server-side techniques are Common Gateway Interface (CGI) scripts, Server Side Includes (SSI), Zope, Lasso, Mason, and Cocoon [14]. An alternative to CGI-based technologies is the Extensible Mark-up language Compiler (XMLC) technology. The compiler is integrated with the open-source Enhydra Application

Server, but is also available separately. We identify three techniques as potential candidates, namely PHP, JSP and XMLC. Our evaluation is summarised in Table 1.

Table 1. Server-side techniques for delivering dynamic content (* = best)**

CRITERIA	PHP	JSP	XMLC
Performance	***	**	**
Database Connectivity	***	***	***
Graphic Format Capabilities	***	***	*
Separation of Content from Presentation	***	**	***
Rapid Development	***	*	*
Cost	***	**	***

In developing Ad-me we have selected PHP. The motivations are as follows:

- High performance: PHP can achieve the highest performance by using PHP caches, such as Zend [3] and APC. PHP caches pre-compile PHP scripts into native CPU code and store them in cache memory.
- Accelerated database connectivity: Database connectivity usually slows down the performance. In PHP, however, connections remain open after a page is finished being processed, thus, dramatically improving the performance.
- Rich set of functions for building images on-the-fly: PHP can deliver dynamically both vector (Flash, SVG) and raster formats (e.g. PNG, JPEG, GIF, and WBMP). This makes PHP a preferable technology for delivering dynamic graphics on mobile devices.
- Separating the web design from logic programming: For large-scale business applications it may be necessary to separate the web design from logic programming. PHP supports template driven designs through such template engines as Smarty. Smarty reads the template files and creates PHP scripts only once, avoiding the need to parse the template files repeatedly.
- Rapid development: The use of libraries such as the HAWHAW PHP class library accelerates the development allowing the developer to deal only with PHP and HAWHAW suppressing the details of each of the constituent mark-up languages. To the best of our knowledge, there is no similar Java library. This library gives the advantage of rapid development compared to traditional approaches (e.g. JSP, XMLC).
- Open-source software: Being open-source software PHP also provides the cheapest option. The only cost for PHP might be for the Zend Cache. However, our experience thus far suggests that use of the free APC cache provide satisfying level of performance.
- Extensibility: PHP is highly extensible and supports Java, XML, COM/DCOM, LDAP, IMAP.

6. Ad-me Functionality

Ad-me (**AD**vertising for the **M**obile **E**-commerce user) investigates the delivery of context sensitive advertising. This functionality is conceived as *push technology* driven by an

underlying user profile and an associated user location. Initial user profiles are used to bootstrap the system and thereafter are dynamically updated based upon user migration and activity within a physical environment. The service is augmented with a range of *pull technology* functions including a *find me function* which will find a desired object that is closest to the users current position. Objects are selected from a list of standard objects including *inter alia* cash machine, taxi rank, pharmacy, train station, and police station. The system addresses interoperability and delivers its service in a manner sensitive to the users technological context.

Inherent in this approach is the need to be able to support certain key user requirements.

- Where am I? Users are localized on a map based interface with the map centered on the user location and the map orientation and user orientation always synchronised;
- Where am I relative to a desired object?
- Guide me to the desired object.
- Tell me about the particular object.
- Book certain resources on behalf of the user, like a taxi, a restaurant table, cinema tickets or theater tickets.

7. System Architecture

The Ad-me system architecture is illustrated by *Figure 1*. We utilise Agent Factory a distributed environment for the rapid prototyping of intelligent agents [19], [21]. Agent Factory is a member of the class of systems that embraces the BDI philosophy. The system offers an integrated toolset that supports the developer in the instantiation of generic *agent structures* that are subsequently utilised by a pre-packaged agent interpreter that delivers the BDI machinery. Specifically a run time environment supports the execution of agents whilst a development environment supports all stages of agent design, specification and debugging. Two central tools are those of an Agent Design Tool which encourages agent reuse and supports the instantiation of generic agent structures, and an agent viewer tool which permits agent state inspection.

The Agent Factory Interpreter is depicted in [10]. The core activity of the Agent Factory system involves a central *perceive-deliberate-act* cycle. Perceptions are registered by the perception builder. These may be invoked through sensory input or conversational input provided by other agents, either of which will typically result in belief revision. The central deductive engine is based on a classic resolution algorithm, which operates under deductive closure generating new beliefs each and every time cycle.

Beliefs and commitments constitute the two fundamental components that govern an agents future directed intentions. Within Agent Factory, two distinct categories of belief exist *current beliefs*, and *temporal beliefs*. Current beliefs represent the beliefs of the agent at a particular point in time, whilst temporal beliefs represent those beliefs an agent holds over some time interval. The syntax of current beliefs consists of standard predicate constructs with the epistemic operator BELIEF. Temporal beliefs augment the syntax of current beliefs with two temporal operators namely *next* and *always*. The *next* temporal operator identifies something that the agent will believe at the next time point. The *always* temporal operator identifies persistent beliefs. Belief rules define belief inferences through logical implications. These remain static for the lifetime of the agent.

Commitment rules express the conditions under which future commitments may be adopted. Typically they take the form of an IF …THEN rule with the condition being comprised of a conjunction of message conditions and mental conditions. If the condition holds at a given instance then the associated commitment is adopted. The Commitment management System (CMS) oversees the adoption and dropping of commitments. Commitments are time stamped and when that time step is reached the commitment is honored.

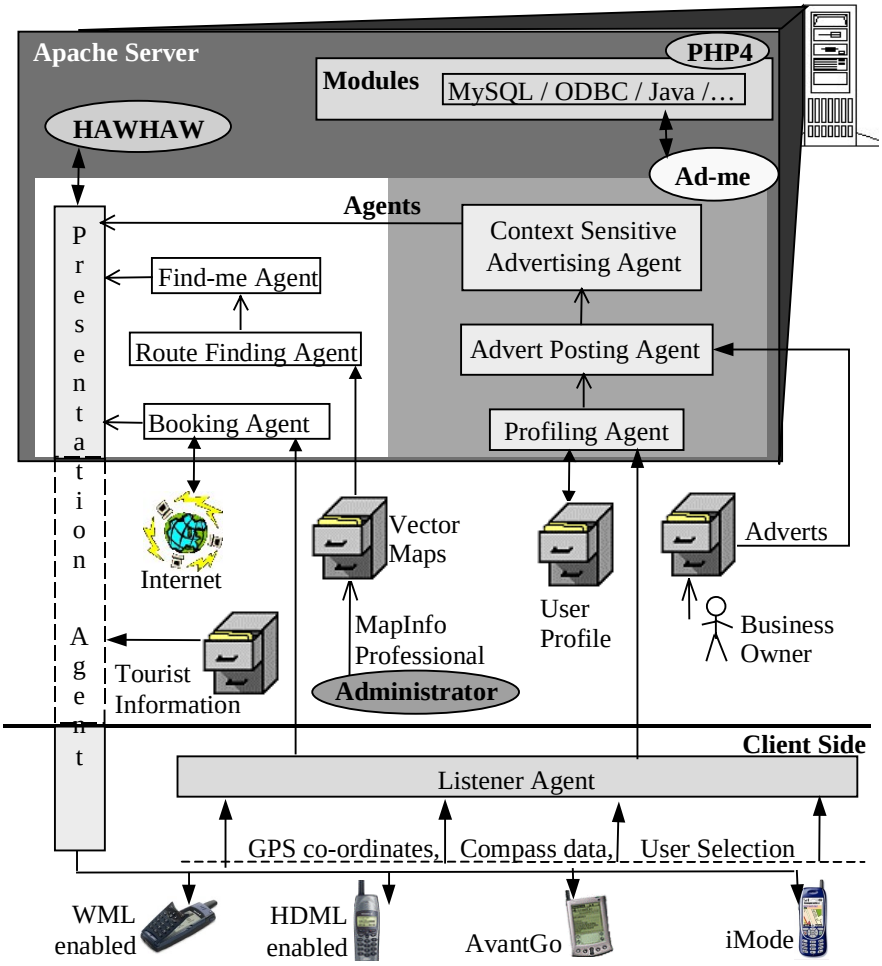

Figure 1: The Ad-me Schematic Architecture

On-going work is developing a lightweight Java agent capability within the Agent Factory framework [17]. Thus heterogeneous agents can co-exist within the Agent Factory system. Lightweight Java agents will reside on the client whilst interacting with heavyweight Smalltalk agents at the server side.

A federation of agents has been delivered which comprise of, Context-Sensitive Advertising Agent, Advert Posting Agent, Profiling Agent, Listener Agent, Find-me Agent, Route Finding Agent, Booking Agent and Presentation Agent. At present not all of these agents are delivered using the Agent Factory machinery. The Listener Agent and the Presentation Agent are conceptually delivered through the PHP system and HAWHAW libraries.

The Find-me Agent is responsible for finding the nearest object of interest, e.g. nearest restaurant, museum, police station, pharmacy and so forth (See *Figures 3 and 5*). The agent uses the result from the Route Finding Agent which queries a GIS database which stores each street as a set of line segments. The Route Finding Agent uses this data together with Dijkstras algorithm and provides the shortest path to a desired object.

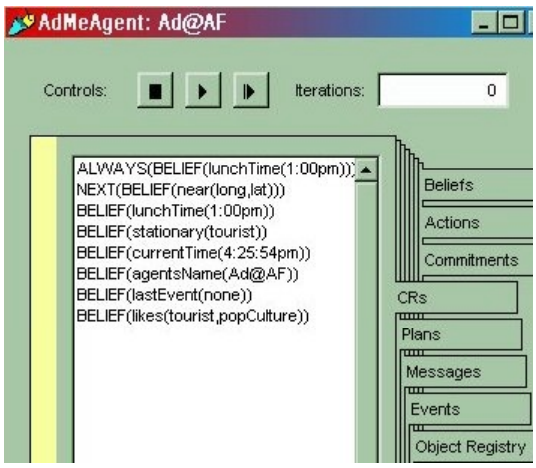

Figure 2: A Typical Belief Set of the Context Sensitive Advertising Agent

Figure 3: iMode Find-me Agent Interface: Object Selection

Figure 4: Context Sensitive Advertising Agent: Advert Details in HTML

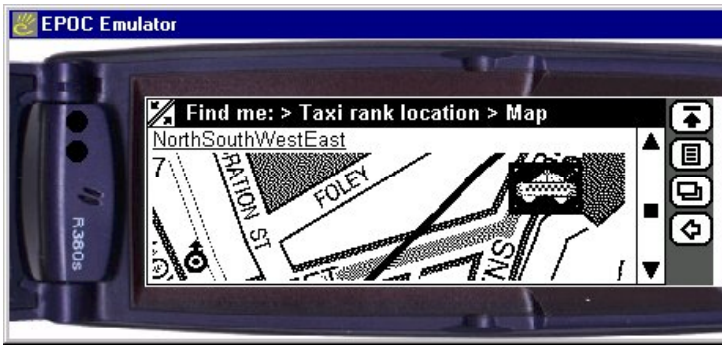

Figure 5: Find-me Agent: Object Display in EPOC

Figure 6: Context Sensitive Advertising Agent: Output in EPOC

The Profiling Agent dynamically maintains a unique user profile. The raw data from which it formulates its decisions are obtained from a listener agent resident upon the client. The Listener Agent monitors and records user location and interface interactions. Currently the browser available on the client's mobile devise fulfils the role of the Listener Agent as well as the Presentation Agent.

The Context-Sensitive Advertising Agent utilises push technology supplying advertisements relative to the user location and perceived need (See *Figures 4 and 6*). The latter is adjudged by interrogation of the profile database. For example, consider our tourist has been wandering for several hours sightseeing, the time nears lunchtime the system thus infers that the user may be disposed to eating. Consequently restaurant advertising would be presented based on their

proximity to the user and user eating preferences and disposable income. *Figure 2* illustrates a typical belief set of the Context Sensitive Advertising Agent. The agent viewer tool is one of the Agent Factory Development Environment toolset and facilitates inspection of the various components that comprise the agent mental state namely; beliefs, actions, commitments, commitment rules, plans, messages, events and the object registry. This tool enables the agent operation to be viewed in real time or simulation mode. The cassette buttons visible support user intervention and allow time slots to be stepped through and the observation of the agent through belief and commitment revision.

The posting of advertisements is the responsibility of the Advert Posting Agent, which receives content and bills the content supplier accordingly. Incremental tariffs give advertisements higher priority. The Booking Agent performs booking on the user's behalf. The Listener Agent will recognise key booking elements activated by the user. For example the reserve table function depicted in *Figure 4*. These events will be communicated to the Booking Agent which will in turn place a booking by either constructing to the appropriate restaurant or venue site. Upon confirmation this will be channelled to the user.

At present interoperability is supported by mapping content to the appropriate viewing technology. *Figure 3* illustrates iMode compatibility whilst *figures 5-6 and 4* illustrate EPOC and Netscape compatibility respectively. The Presentation Agent is responsible for giving the necessary tourist content relevant to places of interest located next to the users location. This agent dynamically generates a map segment image adjacent to the user position. The agent allows the user to "scroll" the map to the North, South, East and West via links. Tourist content relative to map location is subsequently presented. The Presentation Agent presents the content in the appropriate format via HAWHAW. The functionality of the Presentation Agent straddles both the client and server sides.

Our application runs on top of an Apache server version 1.3. The system is implemented using PHP V4.0.5 together with HAWHAW V4.0 library. Databases are delivered via MySQL. User location is obtained by GPS receiver. HAWHAW is a templating tool and consists of various HAWHAW functions, which describe different dialogue elements in the page (such as text, menus, buttons, etc.) and can be called inside a PHP script. The resulting page is then generated in a format depending on the viewer's browser (e.g. WML, HDML, cHTML or HTML). MySQL and ODBC support is built in PHP together with management and maintenance tools. The feature set is complete compared to other databases [8]. The geocoded vector-based data is constructed with MapInfo Professional V5.5. It is our intention to use the APC cache to enhance performance caching compiled PHP scripts.

8. Conclusions and Future Work

Within this paper we have presented the Ad-me system, a context-sensitive advertisement delivery system wrapped in the context of a tourist guide. This system has been partially trailed in Dublin City centre. Our system is innovative in several respects. Firstly, it is one of the first systems that target personalised and location-aware advertisement for cellular phones and PDAs. Secondly, we deploy an agent-oriented approach constructed from a mixture of heavyweight and lightweight agents supporting real-time content retrieval, content presentation and user profiling. Thirdly, the adoption of PHP facilitates interoperability ensuring the delivery of content to a rich

diversity of devices. Future work will undertake larger field trials, extend agent functionality, utilise vector-based maps, develop generic software tools for customisation and localisation of the guide and potentially incorporate the Smarty templating engine.

9. References

[1] ABOWD, G.D., C.G. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton. Cy-berguide: A Mobile Context-Aware Tour Guide. Baltzer/ACM Wireless Networks, 3(5): 421-433, 1997.

[2] ADAMS, N., Gold, R., Schilit, B., Tso, M. & Want, R., An Infrared Network for Mobile Computers, in Proc. USENIX Symposium on Mobile & Location-independent Computing, USENIX Association, Aug. 1993.

[3] BELGE, M., Cimetiere, J.Chr. Zend Cache Performance White Paper. TechMetrix Research, April 2001.

[4] BEUSTER, G., Thomas, B. and Wolff, C., MIA - An Ubiquitous Multi-Agent Web Information System, in Proc. of International ICSC Symposium MAMA'2000, 2000.

[5] BRETZ, E., Perry ,T. X marks the spot, maybe. IEEE Spectrum, April 2000.

[6] BREUGST, M., Choy, L., Hoft, M. and Magedanz, T. Grasshopper – An Agent Platform for Mobile Agent-based Services in Fixed and Mobile Telecommunications, Software Agents for the Future Communication Systems, Springer-Verlag, 1998.

[7] BUSETTA, P. & Ramamohanarao, K., An Architecture for Mobile BDI Agents, Software Agents for the Future Communication Systems, Springer-Verlag, 1998.

[8] CASTAGNETTO, J. et al., Professional PHP Programming, Wrox Press, 1999.

[9] CHEVERST, K., Davies, N., Mitchell K., Friday, A. Experiences of Developing and Deploying a Context-Aware Tourist Guide: The GUIDE Project, in Proc. of MOBICOM'2000, Boston, ACM Press, August, 2000.

[10] COLLIER, R.W., G.M.P.O'Hare, Agent Factory: A Revised Agent Prototyping Environment, in Proc. of AICS-99, University College Cork, Ireland, September, 1999.

[11] ENGE, P. Misra, P. (Editors). Special Issue on Global Positioning System, in Proc. of the IEEE. January 1999.

[12] FROST & Sullivan. The European Global Positioning Systems Market. Strategic Marketing Consulting Report #3352-60, 1998.

[13] HELMER, G., Wong, J., Honavar, V., Miller, L., Lightweight agents for intrusion detection. Submitted to Journal of Systems and Software, 2000.

[14] HINCKLEY, Kee. Web Scripting Tools - Compare and Contrast. November 2000. http://commons.somewhere.com/buzz/2000/scripting.html

[15] JACOBSEN, K., Johansen, D. Ubiquitous Devices United: Enabling Distributed Computing Through Mobile Code, in Proc. of the Symposium on Applied Computing (ACM SAC'99), February 1999.

[16] JOHANSEN, D., Mobile Agent Applicability. in Proc. of the Mobile Agents 1998, Stuttgart, Springer-Verlag LNCS series, September 1998. Also in, Journal of Personal Technologies, Vol 2, No. 2, Springer-Verlag, 1999.

[17] LOWEN, T., O'Hare, G.M.P., Delivering lightweight agents within the Agent Factory framework, Research Report PRISM/5/01, Department of Computer Science, University College Dublin, Dublin, Ireland, 2001.

[18] MARMASSE, N. and Schmadt, C. Location-aware information delivery with Commotion, in Proc. of HUC2000, pp.157-171, Springer-Verlag, 2000.

[19] O'GRADY, M.J., O'Rafferty, R.P., O'Hare. G.M.P. A Tourist-Centric Mechanism for Interacting with the Environment. In Proc. of MANSE'99, Dublin, December 1999.

[20] O'HARE, G.M.P. and Jennings, N.R. (Editors.). Foundations of Distributed Artificial Intelligence, Wiley Interscience Pub., New York, 296 pages. ISBN 0-471-00675,1996.

[21] O'HARE, G.M.P., Collier, R., Conlon, J. and Abbas, S., Agent Factory: An Environment for Constructing and Visualising Agent Communities, in Proc. of AICS98, 1998.

[22] RAPOZA, J., eWEEK Labs builds .asp, .jsp, .php, .cfm versons of e-com site to find best rout to dynamic data, Ocober 30 2000 http://www.eweek.com/a/pcwt0010311/2646051/

[23] WANT, R., Hoper, A. Falcao,V. 7 Gibbons, J., The Active Badge Location System, ACM Transactions on Information Systems 10(1):94-104, 1992.

[24] YOULL, J. Morris, R. Krikorian, P. Maes. Impulse: Location-based Agent Assistance, in Software Demos, in Proc. of the Fourth International Conference on Autonomous Agents. June 2000.