

Aggregation of Composite Location-Aware Services for Mobile Cellular Networks

Alvin Yew¹, Audun Strand², Antonio Liotta¹, George Pavlou¹

Centre for Communication Systems Research
University of Surrey
Guildford, Surrey, GU2 7XH, United Kingdom
{K.Yew, A.Liotta, G.Pavlou}@eim.surrey.ac.uk¹
audun@audunstrand.com²

Abstract. The introduction of context-aware services through service frameworks such as Open Service Access gateways in 3rd Generation networks has coincided with the increasing popularity of Business to Business (B2B) solutions such as Web Services. It is envisioned that B2B characteristics, such as service aggregation will play a part in deploying location-aware services in 3G networks. This paper examines and explores the suitability of service integration models, possible business models, the technical requirements, and suggests a framework for the aggregation and deployment of aggregated composite location-aware services. A prototype of the framework was developed and experiments involving J2EE based and Web Services/SOAP based composite services were conducted and elaborated in the paper. An analysis of the experimental results is presented at the end of the paper.

1. Introduction

The deployment of 3rd Generation (3G) mobile communication networks has provided a wealth of service provisioning possibilities for service providers through new service infrastructures, frameworks, and architectures. Advanced service concepts such as the Virtual Home Environment (VHE) can assist a user in circumventing the difficulties relating to accessibility of personalized services during roaming that were characteristic of past mobile networks generations. A new category of services which can adapt to a user's context, i.e. physical and service operation environment, has been steadily gaining prominence in research. Location-aware services, which can provide service functionality and content relevant to the user's location, belong to that (now often dubbed as '*context-aware services*') category. The discovery of a user's context in 3G networks by context-aware services will be possible through Open Service Access (OSA) and Parlay gateways owned by network operators[1,2]. These gateways encapsulate context determining technologies in the core network such as the Location Service (LCS) for user location determination and other core network capabilities and functionalities[3]. Advances in Business to Business (B2B) service frameworks and middleware have also been progressing steadily over the years as previous state-of-the-art technologies, such as CORBA, have been superseded by more developer-friendly alternatives such as Java 2 Enterprise based architectures, and Web Services using the Simple Object Access Protocol (SOAP)[4]. It is

foreseeable that B2B interactions will have some part in delivering location-aware services to 3G users in the future.

It is common in B2B interactions to combine different, possibly autonomous, services together to form a whole new service. In this paper, we refer to this new service as an *aggregated service*, the individual services that existed before the aggregated service as a *composite service*, and the process of combining the composite services as *service aggregation*.

The study of service composition techniques, requirements, and issues has been identified as a research priority by the World Wireless Research Forum (WWRF)[5]. There is a need to evaluate how service composition may affect the operation and performance of future 3G services such as location-aware services. For example, how do we aggregate composite location-aware services with different requirements (e.g. level of location accuracy)? New possible business models for such scenarios may also be needed and thus require research. This paper aims to tackle and answer the issues concerned with the aggregation of composite location-aware service as well as the characteristics and deployment of aggregated location-aware services.

The format of the paper is as follows: Section 2 introduces the four main models that can be used to integrate composite services. Section 3 analyzes the technical requirements associated with the aggregation of composite location-aware services, and aggregated location-based services operating over mobile cellular networks with a particular emphasis on 3G networks. Section 4 highlights our work on defining a reference framework for aggregating and deploying aggregated location-aware services. It also examines how future business models for 3G networks can have an impact the manner in which our framework is deployed. Section 5 presents an overview of the implemented prototype, details the experiments performed, and evaluates the results of the experiments. The paper concludes by examining the related work done in the area, reiterates the main contributions of our research, and presents potential future work to be done in the area.

2. Overview of Composite Service Integration Models

The development of models for integrating composite services is somewhat similar to that of grid computing. The aggregated service represents the combined service output of all composite services from the user's point of view and is usually in charge of initiating the invocation of the various composite services. Therefore, there are two types of message flows in an integration model; a control-flow and a data-flow. A control-flow is used to control the composite services and can include service initiation, service suspension, and service resumption. A data-flow represents the exchange of service-related and service-specific data, such as localized content, between entities in the model. Figure 1 shows the four main models that are available for integrating composite services[6]. We concentrate our explanations on the first two models as they allow the reader to easily comprehend the operation and purposes of the other two. The four available models for the aggregated service to integrate composite services are:

1. **Centralized Control-flow Centralized Data-flow** – This is the predominant model that is currently documented in literature. The aggregated service is in charge of controlling each composite service, and receives data-flows directly from

them as well. This model is used when it is required to have no dependencies between different composite services, and when the aggregated service requires total control in managing the composite services. A disadvantage of using this model is that there will be an obvious bottleneck at the aggregated service in terms of traffic and processing.

2. **Centralized Control-flow Distributed Data-flow** – This model tries to alleviate some of the bottleneck experienced by the previous model by allowing composite services to exchange data-flow messages between themselves. The model is particularly useful when the output data-flow of a composite service is required as the input data-flow of another composite service, e.g. the aggregated service performs sequential invocation of two or more composite services. This model allows the aggregated service to control a composite service to send its output data-flow directly to the next composite service, which then sends its output data-flow back to the aggregated service, saving both time and network traffic in the process.
3. **Distributed Control-flow Centralized Data-flow** – Distributed Control-flow models are particularly useful when composite service invocations do not need to be sequential. They are also useful when deploying autonomous mobile code such as mobile agents and in active networks. The usage of a centralized data flow in this particular model allows service data integrity as the service data will not be passed between different composite service provider competitors.
4. **Distributed Control-flow Distributed Data-flow** – This model inherits a combination of the characteristics inherent in models 2. and 3.

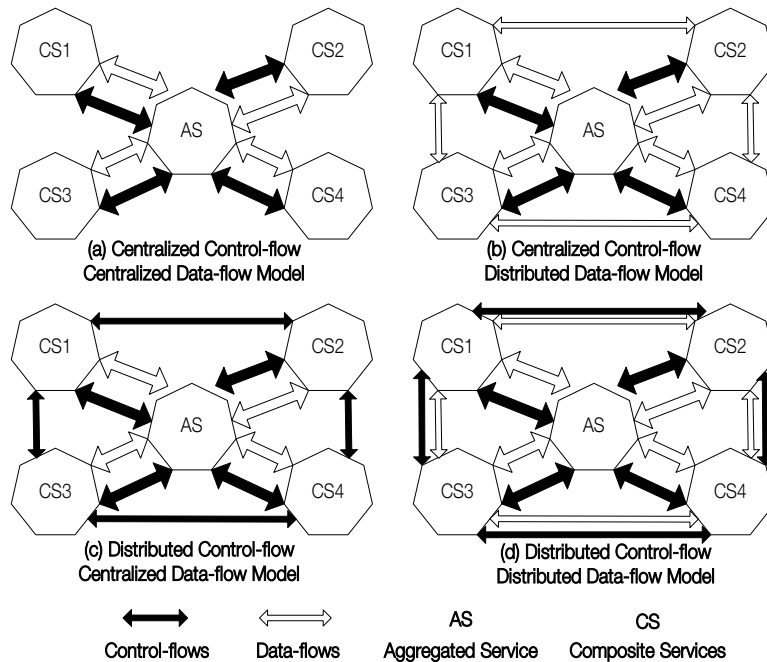


Fig. 1. Integration models for service aggregation

3. Requirements Analysis

3.1 Accommodating Different Types of Aggregated Location-Aware Services

There are three methods of aggregating composite location aware services. Any proposed framework must accommodate these three types of aggregated location-aware services, which are:

1. *Aggregating services of different service categories for a particular location – **CnLl** aggregated service.* For example, an aggregated service is charged with finding a list of restaurants in the area with automated on-line reservation capabilities, a list of car parks in the area including the available capacity at the moment, and a traffic update on the roads in the area as well. This example combines three different independent service categories to provide an aggregated location-aware service.
2. *Aggregating services of the same service category over a range of locations – **CLLn** aggregated service.* For example, a user may want to take a bus from his current location to another location but there is no single bus route or operator that covers this trip. He/She uses an aggregated service that determines his/her current location, contacts different bus operators' timetables and routes to coordinate an interchange within a reasonable transit period, and displays to the user the location of the bus stops to board the bus at each transit point via a map. An additional benefit provided to the user by such a service is that a composite service's *dynamic* information and functionalities, such as bus route detours and cancellations, are incorporated into the aggregated service thus alleviating the user of such concerns. In this situation, the aggregated service may be seen as a single location-aware service whose locality is the union of each composite location-aware service's locality.
3. *Aggregating services of different service categories over a range of locations – **CnLn** aggregated service.* This type of aggregated service combines the characteristics of the previous two. The locality of a *CnLn* aggregated service is also the union of each composite location-aware service's locality.

3.2 Recovering from Limitations of the Overlaid Service Infrastructure

Location-aware services form a service infrastructure overlaid on top of a geographical location. They are, however, different from other service infrastructures because service locality is of primary importance to the functionality of the location-aware service. Figure 2 shows a fictitious service infrastructure available to an aggregated service provider superimposed on a map of London and its surrounding counties. Let us consider that this is a service network of bus operator services in the South-East of England, and that a user wants to travel from central London to Eastbourne for the sake of highlighting our requirements. The following observations can be derived from the figure:

- There is only one bus operator in central London.

- In between central London and the edge of the service network near Eastbourne, the number of competing bus operators in a single location can vary from one to greater than one.
- The service network does not extend to Eastbourne.

It is evident that the aggregated service cannot perform optimally and give the user complete instructions and information on the choice of bus routes from central London to Eastbourne. This is due to the fact that the service network does not extend to Eastbourne. More interestingly, there are two possible reasons for the lack of a service network at Eastbourne. Firstly, there is actually no ‘physical’ bus route to Eastbourne and therefore it is impossible to get to Eastbourne by bus. Secondly, a bus route does exist but the bus operator in that area does not offer its services to the aggregated service provider.

The two possible reasons highlighted in the previous paragraphs implicitly impose a requirement on the transactional nature of aggregating composite location-aware services. A non-transactional ‘best effort’ type of aggregated location-aware service with suboptimal performance can be suitable for *CnL1* aggregated services. For the example in subsection 3.1 of a *CnL1* aggregated service, the user may not mind not knowing the available car parks in the area as long as he is provided the list of restaurants. For the bus operator service discussed earlier in this subsection, which is a *CILn* aggregated service, a transactional approach may be more suitable if it was impossible to get to Eastbourne by bus. However, a ‘best effort’ type of service may be considered if the service was theoretically possible but undeliverable due to a lack of a service network and infrastructure. These possibilities and alternatives highlight the need for the transactional characteristic requirement to be considered on a case to case basis for each deployed aggregated service.

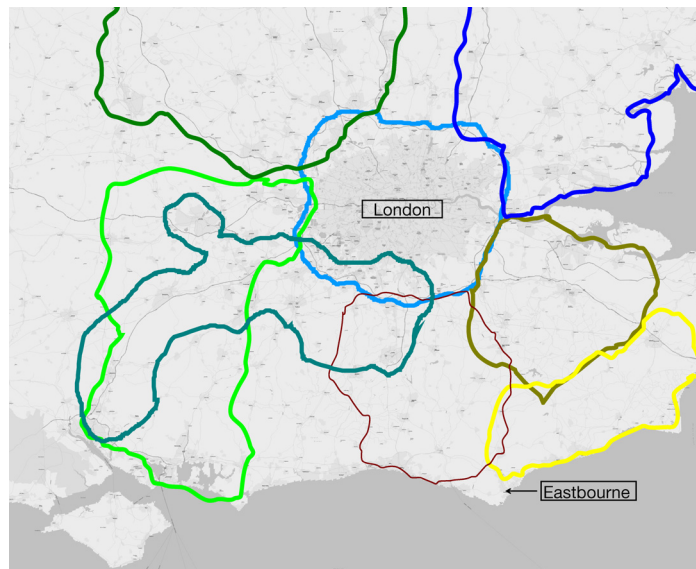


Fig. 2. A service network/infrastructure overlaid on geographical locality

3.3 Accuracy of the User's Location

There are various methods available to ascertain a user's location in current cellular networks. These methods vary widely in the accuracy of determining the location, and the environmental characteristics that are required for a specific positioning method and technology to be available and accurate. The latter usually affects the results of the former in most location-tracking technologies. For example, the Global Positioning System (GPS) requires the user to be mainly outdoors so as to receive a Line of Sight (LOS) signal from the Low Earth Orbit (LEO) satellites. Location accuracy when using GPS to determine a user's location increases as the number of satellites over the user's location increases. While many positioning methods and technologies do exist, e.g. Round-Trip Time (RTT), Angle of Arrival (AOA), Reference Node Based Positioning (RNBP), 3GPP has standardized three methods for the Location Services (LCS) functionality in 3G mobile networks. They are Cell ID based positioning, Observed Time Difference of Arrival (OTDOA) positioning (which is based on Time Difference of Arrival positioning), and Assisted GPS positioning (which essentially is GPS positioning). Figure 3 shows the difference in accuracy for these methods and technologies in a variety of operating environments[7]. OTDOA works well indoors and provide reasonable accuracy when more than two neighboring base stations are used in determining the user's location. GPS is the most accurate but requires LOS between the user and three or more satellites to perform well. Cell ID based positioning is the least accurate and its accuracy is inversely proportionate to the size of the cell that the user is in.

It can be easily deduced that the accuracy of the user's location is a major factor when provisioning location-aware services. Inaccurate location positioning of the user can result in providing the user with services that he/she does not want or need, thus decreasing user satisfaction and service reputability. The importance of this factor increases when various location-aware services are aggregated together. Although all aggregated services would suffer to some extent from positioning inaccuracy, a *CnLI* aggregated service would be more likely to be affected than a *CILn* or a *CnLn* aggregated service as its overall service locality is smaller than that of the other two (assuming that the area of each composite service locality is equal and exclusive) – a small inaccuracy would constitute a greater percentage of error over a small area than a large area. However, this may not be the case for an aggregated service with functionalities that has strict positioning accuracy requirements regardless of aggregation type, e.g. a service involving the location of the nearest hospital or 24-hours emergency medical clinic. Therefore, the requirement for positioning accuracy from an aggregated service depends on both the type of aggregated service, and the aggregated service functionality.

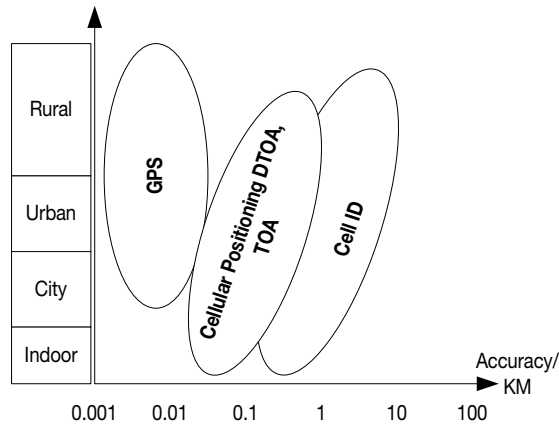


Fig. 3. Accuracy of various key positioning techniques and technologies

4. Framework for Deploying Aggregated Location-Aware Services

4.1 Design Considerations and Features

Figure 4 gives a functional overview of our proposed framework for accommodating aggregated location-aware services using composite services. We stress that the proposed framework should not be seen as a ‘one solution fits all’ approach for deploying such services as we disregard other significant issues and functionalities associated with location-aware services such as content adaptation to terminal capabilities, accounting, user authentication and session management features. These issues and features are pertinent to most service frameworks, and we concentrate on issues that we considered to be specific to the aggregation of location-aware services in this paper.

A huge factor in the design of the framework is that it must fit within a sensible and realistic business model. During the course of our research leading to this paper, we visited a major UK network operator, as well as an independent investment banker, for consultations on future viable business models for deploying location-aware services in 3G mobile communication service frameworks. The framework design accounts for a key point that they raised during our meetings – information about a user’s location is extremely private, and will not be released to any service provider that requests for it through the OSA interfaces. The access to such information through OSA is negotiated through off-line service level agreements, and they predicted that only a few privileged service providers will be considered trustworthy and reliable enough to gain such access. The network provider ultimately has a responsibility to its customers to protect their privacy as well as its own reputability.

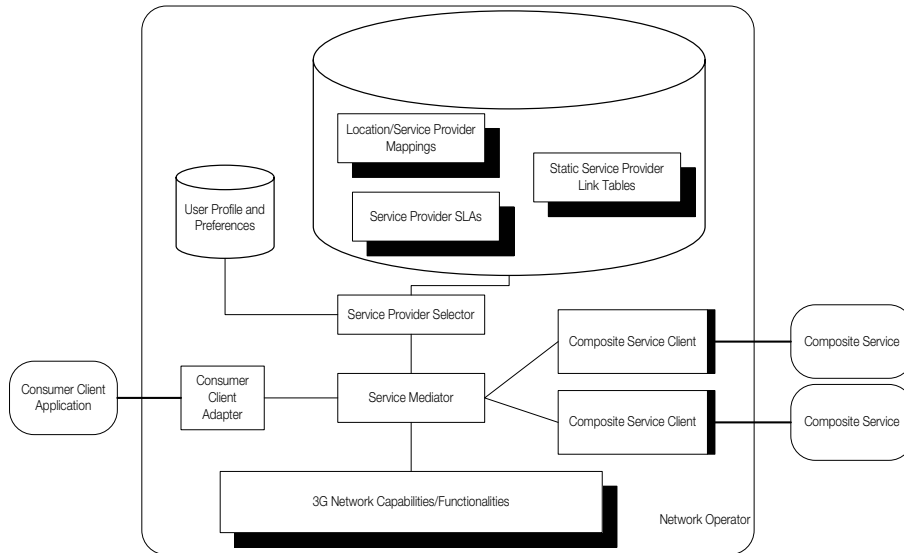


Fig. 4. High-level functional view of the framework

Another point that the UK network operator brought up was the oft used term ‘*value-added service*’. According to their opinion, value-added services are those that the network operators cannot implement themselves due to reasons such as proprietary technologies, i.e. patented, and copyrighted content. They also stated that most other service functionalities would be created in-house if possible. This closed behavior can also be attributed to the business model success of i-mode in Japan, where network operators were controlling, monitoring and coordinating the services and content available to their users so as ensure high usability and user satisfaction. These factors imply that service aggregation would most probably be conducted within the network operator for 3G networks in the future. Therefore, we do not include interactions with an OSA/Parlay gateway in figure 4, as our framework places the emphasis on the network operator to initiate and control the service aggregation process. We do, however, acknowledge the possibility that composite services may also be aggregated services themselves - i.e. they use other composite services to perform and operate their services. By placing the service aggregation process within the responsibilities of the network operator, it is also possible to conceal the location of a specific user from composite location-aware services. The latter would not require the user’s identity for billing or authentication as the network operator would be charging the user on the composite service provider’s behalf. A network operator controlling the service aggregation process also implies ease of control over the choice of positioning methods and technologies used in ascertaining the user’s location, thus providing some flexibility in satisfying requirements on positioning accuracy from various aggregated location-dependent services. It is, however, still possible to deploy our framework within the domain of a value-added service provider (to a 3G network operator). This would mean interpreting the interface

between the *service mediator* and the *3G network capabilities/functionalities* functional components to be that of OSA's (refer to figure 4).

We also designed the framework to incorporate flexibility in as many aspects of service provisioning and management as possible, and therefore the following features are present in the framework:

- The framework interoperates with a range of service deployment technologies available such as Enterprise Java Beans, Web Services/SOAP etc., allowing composite service developers flexibility in choosing service deployment middleware technologies.
- The framework allows more than one service access point to the aggregated service so as provide flexibility in service delivery to the service consumer. E.g. Java 2ME midlets with Java RMI clients, SOAP clients, HTTP-XHTML/HTML/WML.
- The framework design allows service aggregation using any of the four service integration models presented in section 2.

4.2 High-level Component Description

This subsection explains the functionalities of each major component in the framework as shown in figure 4.

- *Composite Service Client*: This component serves two purposes. Firstly, it acts as an adapter (design pattern) between the composite service and the *service mediator*, allowing the possibility of heterogeneous service deployment technologies between the two[8]. E.g. Java EJB communicating to a SOAP agent. It also acts as a facade (design pattern) by providing a consistent control and service data model to the *service mediator*[8] It does this through adapting the control and service data to match that expected by the composite service.
- *Service Mediator*: The service mediator contains the core logic of the service aggregation process. It acts as a proxy (design pattern) for all the other significant components in the framework[8]. Its main responsibilities are to control and collect service data from the composite services via the *composite service client*, determine and request the appropriate positioning technique used in the 3G network based on the location accuracy required, construct and aggregate the composite services, and deliver the service to the *consumer client application* via the *consumer client adapter*. It is assisted in the aggregation process by the *service provider selector* component.
- *Consumer Client Adapter*: This component acts as an adapter between the consumer client application and the service mediator. It receives the aggregated service data from the service mediator, transforms it to the correct presentation format if necessary (e.g. HTML, WML etc.), and delivers it to the *consumer client application* in a protocol that the latter understands (e.g. HTTP/SOAP etc). It is acts as a service access point by allowing the consumer application client to request and initiate the aggregated service.
- *Location/Service Provider Mappings*: This is actually a database table that provides a list of Service Providers within a geographical locality (e.g. Radio Access Network Cell, City, Town etc.) for a specific service category. A table entry also states if a service is not available in a locality because of a lack of service providers or it is theoretically impossible to provide the service.

- *Static Service Provider Link Tables*: This is actually a database table that represents a service network/infrastructure for a specific service category in tabular form. Table entries include the number of ‘hops’ required for a service provider to get to each other service provider within the service network. This table will be used mainly for selecting a list of available service providers for *CILn* and *CnLn* aggregated services. It is also used to determine if the service network/infrastructure can support the required service, and thus influence the operation of both transactional and ‘best effort’ aggregated services.
- *Service Provider SLAs*: Service Level Agreements (SLA) are a key part of service management and deployment. They contain information about Acceptable Service Response Time, Guaranteed Network and Application QoS between the aggregated service and the composite service, cost of using the service, Unit of Work definitions etc.[9,10,11]. They are used in the framework to ensure that the performance of any single composite service does not hinder the overall performance of the aggregated service.
- *User Profile and Preferences*: The user profile and preferences houses a list of service specific user preferences, such as cost of using the aggregated service (this can be partly determined from the Service Provider SLAs), location accuracy and response time, for a particular aggregated service.
- *Service Provider Selector*: This component contains the logic of selecting multiple service providers that will form the aggregated service according to the user preferences, and other aggregated service requirements discussed earlier.

5. Implementation of an Aggregated Road Traffic Router Service

We implemented our framework on Sun Microsystems’s Sun ONE Application Server 7 on an Intel Pentium III 1.0 GHz server and a Sun PointBase database server. Most of the components in the framework were written in Java, and the *service mediator* and *service provider selector* components were implemented as session EJBs. The LCS in the 3G core network was simulated by a Java application triggering location information using the data model specified in [12]. We decided on, and developed, a road traffic router aggregated service to perform our experiments on the framework. This aggregated service provides a user with a choice of routes from the current location to his/her stipulated destination according to the user’s preference (e.g. distance, journey duration etc.). Every composite service involved in constructing the aggregated service, e.g. a private toll-paying highway company, provides the routing functionality and traffic monitoring functionality for its own locale in the journey. The road traffic conditions within different locales can influence the routing between the user’s embarkation and destination points, and all of the composite services collectively provide a *CILn* aggregated service via a Centralized Control-flow Centralized Data-flow integration model. We chose road traffic router service because the UK Highway Agency and the Transport Research Laboratory were researching on providing a similar service in the UK[13]. A traffic monitoring system also constitutes as a valid value-added service as the monitored roads may be owned by a private company, e.g toll-based roads. Furthermore, our framework can be easily adapted into their business model, thus conceptualizing such a service is

realistic and non-idealistic. The following subsection explains the choice of the road traffic routing algorithm used in the composite service for our experiments.

5.1 Choice of Road Traffic Routing Algorithm

The tree model is a good possible representation of a road infrastructure. It is also used in other domains such as data networks. This means that there is a lot of research in this field, and algorithms that one can use in finding the desired paths in such a tree. There are some intricacies in mapping such a tree representation to a road reality. Traffic rules must be obeyed and that means that there are one way roads, and also intersections where some turns are prohibited. This can be solved by introducing extra links so that the link is directional [14]. The cost of the links between the nodes can be a variety of metrics; for road traffic the cost can be distance, travel time, safety etc [15]. To facilitate the usability, the data model must also consist of secondary information such as road names etc. so that the user can get the information needed to find the desired route. For instance, the Danish Road Directorate maintains more than 100 attributes for each road in their database. [16]

A non-adaptive routing approach has its advantages in the ease of implementation and its simplicity. It does not have an initial cost, and does only require processing effort when needed. When used on a dynamic data model it is capable of delivering dynamic routing, in the sense that it always uses the present network topology as a base for the routing. There are many algorithms that one can use in the non-adaptive approach; [14] lists two, Dijkstra's algorithm and Lee's algorithm, while [15] and [17] focuses on Dijkstra's algorithm.

Lee's algorithm comes from track routing on printed circuit boards. The method simulates an inkblot spreading out over a piece of paper, centered on the starting point. Fawcett & Robinson chooses this algorithm since it always knows the cost from the origin node to the present node [14]. Since their system takes a statistical approach to cost calculation where they know the cost of roads at different times of they, this is vital for them. In this application this is not important since the routing will be done real time, on a dynamic data model.

The Dijkstra's shortest path algorithm guarantees to find the best route (lowest cost) in a network, if such a route exists. The algorithm considers links with smallest cost first, and when a node is reached all links from this node is inserted in a priority queue, based on the cost it represents. Every link is then retrieved from the priority queue with the smallest value first. In this manner the whole network is visited until the destination is reached. When the quickest route is found, the algorithm tracks back to record the route of the shortest path. The cost of this algorithm is $O(v^2)$ for a network with v links. The hierarchical nature of the data model will keep the number of nodes relatively small and help maintain acceptable performance.

The dynamic nature of this application is important. This means that if changes in the network topology mean that the route given to the client is no longer the shortest, then this information must be sent to the client together with a new improved route. We have considered all of the above algorithms for finding the shortest path in a network for our road traffic router service. Adaptive routing has a short computation time, but requires lots of resources and has high implementation complexity. Lee's algorithm has some advantages, but they are not very relevant in this application. Dijkstra's algorithm is relatively easy to implement, and should with the

improvements described be fairly quick, especially with the hierarchical nature of the data model in mind. Gutman states that when altering an algorithm to take domain specific knowledge into account, one can improve the performance of the algorithm. The routing application in the composite traffic router service used an algorithm based on Dijkstra's algorithm, with the improvements and modifications detailed by Gutman [17], and incorporate distance, average speed, and journey duration as road cost attributes.

5.2 Analysis of Prototype Experiments

We decided to examine two deployment scenarios in aggregating the road traffic composite services by implementing all the composite services as EJBs, and as Web Services (SOAP). To build the Web Services version, we used Sun Microsystems' Java Web Services Developer's Pack v1.1, and its JAX-RPC APIs. For the EJB version of the service, we used the Sun ONE Application Server 7 and the associated J2EE 1.3 APIs. Method calls between EJBs were performed through Java's Remote Method Invocation over Internet Inter-ORB Protocol (RMI/IIOP). We performed all our experiments on a single computer as there were no intentions in our experiments to analyze delays caused by network traffic and latency due to network QoS issues. Although the size of the network traffic from control and data SOAP messages for Web Services depends on the design of the Web Service interface, the service data model, and the schema used, we designed them to be similar to those in the EJB scenario in our experiments for comparison purposes. In our experiments, we were more interested on how the choice of a service middleware platform for composite services, e.g Web Services and J2EE, can affect the service response time of the aggregated service, which can be due to the traffic size of the control and data messages, without the effects of delays in the network. We scrutinize the level of response times because it is a metric that will explicitly affect the user's enjoyment of the aggregated service. When performing the experiment, we assume that a service level agreement between the aggregated service provider and the composite service provider has been pre-established, and the relevant data models have been agreed on. We also developed the respective *composite service client* components (an EJB and a JAX-RPC/SOAP version) for each scenario. To complete the entire scenario, we developed both a Java RMI/IIOP based, and a Java 2ME midlet with SOAP agent based *consumer client application* each for the EJB and Web Services scenarios respectively. We measured the response times using appropriate time stamps in the programming code and for each scenario, we invoked the aggregated service for five different routes of similar complexity involving three separate composite service providers with different service locality area size. We measured the response time for each route 10 times, excluding the discarded first run – mandatory when testing Java code reliably (a practice attributed to the idiosyncrasies of the Java Virtual Machine). Table 1 shows the response times measured in milliseconds.

Table 1. Response times of the experiments (milliseconds)

	Client		Aggregated Service		Composite Services Total		Client Overheads		Aggregated Service Overheads	
	WS	EJB	WS	EJB	WS	EJB	WS	EJB	WS	EJB
Service Architecture										
Route 1	700	802	423	431	174	303	277	372	249	128
Route 2	715	679	407	346	186	247	308	333	221	99
Route 3	510	560	242	296	130	181	267	264	112	115
Route 4	478	701	242	415	108	281	237	286	134	134
Route 5	482	463	270	205	117	99	212	258	153	106
Average	577	641	317	338	143	222	260	303	174	116

WS = Web Service Scenario EJB = Enterprise Java Beans Scenario

The ‘Composite Services Total’ column in table 1 represents the total time taken for all the composite services to compute an optimum traffic route in its locality without including the time needed to serialize, un-serialize, parse, and un-parse data and control messages. The ‘Aggregated Service’ column shows the time elapsed between the *consumer client adapter* in the framework initiating the aggregation process through the *service mediator*, and the *consumer client adapter* having the aggregated service data ready for transmission back to the *consumer client application*. This too does not measure the time required to serialize, un-serialize, parse, and un-parse data and control messages. The ‘Client’ column shows the total time taken in milliseconds from the sending of the aggregated service request to the receipt of the aggregated service response (inclusive of processing all control and data messages) by the *consumer client application*. The ‘Client Overheads’ column represents the time taken to serialize, un-serialize, parse, and un-parse control and data messages between the *consumer client application* and the *consumer client adapter*. The ‘Aggregated Service Overheads’ column shows the total time taken to serialize, un-serialize, parse, un-parse control and data messages between the *composite service clients* and the associated composite services. The interesting result in table 1 is the comparison between the web service and the EJB scenarios for ‘Aggregated Service Overheads’. This is mainly the overheads incurred when collaborating with the different composite services. We can see that EJB’s RMI/IIOP incurs lower overheads than the Web Service’s JAX-RPC/SOAP on the average. This is because when objects and value objects are serialized in RMI/IIOP, they are also compressed before being sent through the network, whereas JAX-RPC/SOAP messages are not compressed. Therefore, the size of RMI/IIOP messages sent and received is less than that of JAX-RPC/SOAP. Furthermore, the parsing and un-parsing of SOAP’s XML based messages require more processing power and time than Java’s RMI/IIOP.

6. Related Work and Conclusions

Though there are many documented service middleware and frameworks developed for location-aware services, our work is most closely related to the IST MOBIVAS project which was a recent EU funded project[18]. Their middleware resides on top of an OSA/Parlay gateway and caters primarily to providing a full service support platform for context-aware value-added service providers. However, their platform differs from our framework by not considering issues concerning service aggregation, and the significance of location accuracy in service performance. [19,20] both cover platforms for service aggregation that emphasizes on dynamicity. These platforms are extremely suitable for on-the-fly B2B transactions and on-demand service SLAs. Our research differs in approach as our business model assumes that B2B relationships and SLAs are negotiated off-line.

The standardization and impending implementation of the LCS technology in 3G mobile communications is indicative that location-aware services will soon play a significant role in the design of future service architectures and frameworks. As the influx of B2B oriented service architectures such .Net and Web Services grows, the integration and aggregation of composite location-aware services will seem evermore attractive to service providers. In this paper, we attempt to identify the main hurdles that aggregated location-aware services will face and suggest a reference framework based on our informed and calculated prediction on the future business model for such services. Nevertheless, our framework can also operate on top of an OSA/Parlay gateway if required.

We identified the three main types of aggregated location-based services, and examined their requirements and possible dependencies on location accuracy and transactional behavior. We have also presented on the various service integration models available for aggregating location-aware services, their benefits pertaining to location-aware services, and ensured that our framework allow flexibility in deploying any of the models available. The experiments conducted with our framework highlighted the subtle differences when aggregating composite services using two well received service middleware architectures: the current predominant favorite J2EE Enterprise Java Beans, and the up and coming Web Services/SOAP based architecture.

There are two certain avenues in which we would further pursue the research detailed in this paper. It would be interesting to perform a more detailed analysis on the size of the control and data messages, so as to find the correlation between the time lost on serialization/parsing overheads and the control and data message size. Another intriguing experiment that we have planned for the near future is to examine the performance of our framework in a Centralized Control-flow Distributed Data-flow integration model, as we feel that this may yield some gains in response times.

References

- [1] 3GPP TS 29.198, Open Service Access (OSA); Application Programming Interface (API); Part 1-12, 3rd Generation Partnership Project.
- [2] Parlay Group, "Parlay API Specifications 4.0", <http://www.parlay.org/specs/index.asp>, (current May 2003).
- [3] 3GPP TS 22.071, Location Services (LCS); Service Description Stage 1, 3rd Generation Partnership Project.
- [4] M. Champion, C. Ferris, E. Newcomer, D. Orchard, "Web Services Architecture", W3C working draft dated 14 Nov. 2002, <http://www.w3.org/TR/2002/WD-ws-arch-20021114/> (current May 2003).
- [5] R. Popescu-Zeletin et al, "Service architectures for the wireless world", *Computer Comms.*, vol. 26, no. 1, Jan. 2003, pp. 19-25.
- [6] D. Liu, K.H. Law, G. Wiederhold, "Analysis of Integration Model for Service Composition", Proc. Wkshp on Software and Performance (WOSP'02), ACM Press, Rome, Italy, 2002, pp. 158-65.
- [7] H. Kaaranen et al, UMTS Networks Architecture Mobility and Services, John Wiley & Sons, Chichester, England, 2001, pp. 161.
- [8] E. Gamma et al, Design Patterns: elements of reusable object-oriented software, Addison-Wesley, Reading, Mass, 1994, pp. 139-50, 185-94, 207-22.
- [9] L. Lewis, P. Ray, "On the Migration from Enterprise Management to Integrated Service Level Management", IEEE Network, Jan-Feb 2002, pp. 8-14.
- [10] Keller, A. Koppel, K. Schopmeyer, "Measuring Application Response Times with the CIM Metric Model", Proc. 13th IFIP/IEEE Int'l Wkshp Distributed Sys (DSOM 2002), Montreal, Canada, 2002, pp. 66-81.
- [11] M. Debusmann, A. Keller, "SLA-Driven Management of Distributed Systems using the Common Information Model", Proc. 8th IFIP/IEEE Int'l Symp. on Integrated Network Mgmt (IM 2003), IEEE Press, Colorado Springs, CO, 2003, pp. 563-76.
- [12] ETSI ES 202 915-6v1.1.1, Open Service Access (OSA) API; Part 6: Mobility SCF, Jan. 2003, European Telecommunications Standards Institute.
- [13] M. Yearworth et al, "A CORBA Service for Road Traffic Information on the Internet", Proc. Int'l Symp Dist.Objects and Applications (DOA'00), IEEE CS Press, Antwerp, Belgium, 2000, pp. 231-41
- [14] J. Fawcett, P. Robinson, "Adaptive Routing for Road Traffic", IEEE Computer Graphics and Apps., May/Jun. 2000, pp. 46-53.
- [15] P. Ioannou et al, "Modeling and Route Guidance of Trucks in Metropolitan Areas", University of Southern California, Center for Advanced Transportation Technologies Internal Report, <http://www.usc.edu/dept/ee/catt/reports/mettrans/>, (current May 2003)
- [16] C.S. Jensen, "Research Challenges in Location-Enabled M-Services", 3rd Int'l Conf Mobile Data Mgmt, 2002, pp.3
- [17] R. Gutman, "Priority Queues for Motorists", Dr. Dobb's Jrnl, Sept. 2002, pp. 89-92.
- [18] "MOBIVAS: Downloadable Mobile Value-Added Services through Software Radio and Switching Integrated Platforms", <http://mobivas.cnl.di.uoa.gr>, (current May 2003)
- [19] G. Piccinelli, G. Di Vitantonio, L. Mokrushin, "Dynamic service aggregation in electronic marketplaces", Computer Networks, vol. 37, no. 2, Oct. 2001, pp. 95-109.
- [20] D-R Liu, M. Shen, C-T Liao, "Designing a composite e-service platform with recommendation function", Computer Standards & Interfaces, vol. 25, issue 2, May 2003, pp. 103-117.