ABSTRACT
The practical success of the future generation mobile systems such as 4G relies largely on its flexibility in providing adaptive and cost-effective services. Service discovery is an essential mechanism to achieve this goal. Based on the investigation of the existing service discovery protocols, this paper proposes a new service discovery methodology for future generation mobile systems: model-based service discovery or MBSD. MBSD takes advantage of the OMG (Object Management Group) MDA (Model-Driven Architecture) technique. The system architecture of MBSD and its operation are presented and implemented. The proposed methodology is validated via a mobile service scenario.

Categories and Subject Descriptors
D.2.2 [Design Tools and Techniques]: Object-oriented design methods, System Design.
H.1.1 [Systems and Information Theory]: General systems theory, Information theory, Value of information

General Terms
Algorithms, Design, Experimentation

Keywords
Service Discovery, Model-Driven Architecture, Wireless Mobile Networks

1. INTRODUCTION
Recent years have seen a significant increase in research and development of mobile and wireless networks, including UMTS, IEEE 802.11, mobile ad hoc networks, sensor networks, etc. The integration of different kinds of wireless mobile networks is also emerging. The future success of these mobile systems lie in its ability to provide users with cost-effective services that have the potential to run anywhere, anytime and on any device without (or with little) user attention. Services of these features are usually termed as pervasive services, which is an active branch of pervasive (or ubiquitous) computing [1]. A pervasive service can be a simple one such as helping a user on a mobile device (such as a PDA or a smart phone) find his/her favourite restaurant in the vicinity based on his/her current location. However, it is more of a composite of the abovementioned small services, for example, a Virtual Office Environment (VOE) service [2] that could enable Julia on a PDA to continue her normal office work such as dealing with emails, editing meeting memos left on her office desktop even when she is at the Heathrow Airport waiting for boarding. So service adaptability is an important characteristic of pervasive services. This paper focuses on a specific enabling technique to implement service adaptability: service discovery.

Service discovery is an essential mechanism broadly utilized by researchers and competing industries to pursue dynamic and automatic software system composition, configuration and adaptation. It impacts on the cost-effectiveness and efficiency of the creation, operation and maintenance of pervasive services of big network operators and service providers such as BT and its 21CN (21st Century Networks). However, the current situation of service discovery to some extent hinders the service supporting systems from pursuing their maximum effectiveness. Existing service discovery mechanisms have adopted various service description languages, thus limiting the scope of global service discovery. Though each of them has its advantages and disadvantages in terms of expressiveness and query efficiency, they are primarily bound to the architecture of the individual discovery mechanism. Most importantly, the current service discovery mechanisms, either it being for wired networks such as UPnP [3] or for wireless ad hoc networks such as Konark [4], all fail to provide incentive for service composition. In contrast, the discovered service components as a result of any service discovery will typically be used to compose new services (including integrating with the current user services). It is natural to have a service discovery mechanism that inherently provides such insight so as to ease the difficulty of service composition.

Motivated primarily by the above considerations, this paper introduces a novel methodology for service discovery, i.e., model-based service discovery (MBSD). A model-based service discovery approach coincides with the emerging trend of modern software engineering, i.e., model-driven computing with OMG MDA (Model-driven Architecture) [5] as a strong representative. The MBSD protocol makes use of XML as its description
language, to allow more expressiveness for service description and more powerful matching for service discovery, as is the case with WSDL (Web Service Description Language). Rather than manually creating service description as in WSDL, the XML-based service description is automatically generated in MBSD protocol from models of services. These models have existed as a MDA model-driven component-based software system engineering method has been employed by an increasing number of service providers to develop their services and service supporting systems. MBSD embraces future technologies for software development.

In the light of these challenges, this paper first discusses the model-based service discovery methodology by looking into the related research domains one by one. Then, after presenting the service description language used in MBSD, the full system architecture of MBSD and its service discovery procedure are represented. Finally a prototype MBSD system is demonstrated along with a brief presentation of a scenario-based validation.

2. A METHODOLOGY FOR MODEL-BASED SERVICE DISCOVERY FOR FUTURE GENERATION MOBILE SYSTEMS

2.1 Service Discovery

In this sub-section, two main service discovery techniques, Jini and UPnP, initiated for wired networks are discussed first, followed by two others, DEAPspace and Konark, for wireless networks. Some features of MBSD are presented in general comparison with these service discovery mechanisms.

- Jini: Jini is a Java-based technology introduced by Sun Microsystems. Three protocols works together to constitute the core of Jini: discovery, join and lookup [3]. It uses a multicast-based advertisement to announce its existence to the potential service clients. The service discovery is based on unicast after knowing the presence of service brokers.
- UPnP (Universal Plug and Play): the UPnP Forum, headed by Microsoft, is in charge of the standard’s developments [3]. UPnP involves advertisement, discovery, and control of networked devices, services. Service description in UPnP is based XML and control messages are expressed as a collection of SOAP objects and their URLs in the XML file.
- DEAPspace: DEAPspace, developed by IBM [6], evolves the service discovery from wired networks to wireless networks (i.e., single-hop mobile ad hoc networks). Each node equipped with DEAPspace maintains a view of all the services present in the network via periodical exchange of service information. Service discovery is carried out by a broadcast of service request message to a requester’s neighbours. In DEAPspace, broadcast is scheduled sooner than usual in order to obtain prompt responsiveness and accuracy in service discovery.
- Konark: designed also for wireless networks, Konark [4] furthers DEAPspace to multi-hop wireless ad hoc networks. Konark proposes a more efficient way for service information propagation. Each device in Konark community has a Konark SDP (Service Discovery Protocol) Manager that discovers required services on behalf of Konark applications and advertises the service information provided by the device. Konark also specifies a service description language using XML. Konark, like DEAPspace, utilizes caching of service information on each node to improve service discovery efficiency.
- MBSD proposed in this paper adopts a fully distributed architecture over multi-hop wireless networks, i.e., each mobile node is a service provider and is its own service broker. However MBSD uses a combination of caching mechanisms and limited broadcast to further improve its efficiency. Furthermore, the content structure of its service registry is well organized to reflect the domain-specific feature of mobile applications. There is no need for service advertisement (i.e., limited broadcast) in service advertisement. XML is employed by MBSD to describe the service. However, rather than creating these description manually, MBSD makes re-use of the service information already existing in the service models.

2.2 Model-driven Approach

The Model-Driven Architecture (MDA) [5] is an approach to IT system development fostered by the OMG. Its essence is the concept of separation between the specification of a system’s essential functions as a Platform Independent Model (PIM) and the realisation of the system using more specific and detailed platform as Platform Specific Model (PSM). Its promising platform-, language- and middleware-neutral features brings great impact on the current methods and techniques applied to manage software development process.

Apart from IBM Rational Rose and its UML (Unified Modelling Language), other more specific MDA tools have made their appearance in the market, such as XMF toolkit from Xactium Ltd. [7], Open Source Alliance Eclips also have several initiatives targeting MDA toolkit, such as EMF (Eclips Meta-model Facility). MDA is regarded as a very promising future software development methodology. The software engineering community has seen an increasing number of software components being developed in MDA manner, including numerous activities in applying MDA for service engineering [8], and trialling MDA in OSS (Operational Support Systems) operated by network and service providers such as BT.

In MDA family, MOF (Meta-object Facility) is used for describing meta-data, which can be represented, for instance, by UML. In pervasive service engineering, meta-information about services is utilized for service discovery. Therefore, rather than re-inventing metadata from services, we could directly use the existing MOF metadata that describe the features (typically via interface description) of service for the benefit of service discovery. Since in MBSD service description is based on XML (more later on), a mechanism that transforms any MOF-based meta-model into an XML format is needed. This is where XMI (XML Metadata Interchange) [5] comes into place. XML is the official OMG specification for exchanging model information between modelling tools or model repositories. XMI uses XML as its carrier. Furthermore, XMI can be read by other modelling tools which then regenerate MDA models of the service based the XMI.

2.3 Future Generation Mobile Systems

Diverse network structures and protocols cause lots of problems in network integration and service integration. For a more efficient utilization of variable network materials, two or more
network structures are integrated. For instance, iCAR [9] tries to introduce mobile ad-hoc networks into a cellular network to improve the latter’s call blocking rate in hot spots such as sporting venues or on the scene of emergency events. The heterogeneity in network structures is a typical feature of 4G mobile networks. In recent years, small portable devices have been increasingly equipped with multiple communication interfaces. Some new multiple-interface servers are also under design and development. These facilities make the integration of different networks feasible and more convenient. Therefore, heterogeneous networks will become the main part of future mobile systems and offer people more flexible network services.

The service discovery work to be discussed in this paper is based on the heterogeneous network depicted in Figure 1, which is currently being researched in the University of Essex. This is a similar network structure to that of iCAR [9]. Two air interfaces are utilized for the communication between nodes: C (Cellular) interface that operates at a cellular network frequency (in-band), and A (Ad-hoc) interface that operates at an ad-hoc network frequency (out-of-band, e.g. IEEE 802.11).

![Figure 1: Mobile Network Physical Structure](image)

Base Station (BS) is same as present cellular network base stations with C-interface. A base station uses its C-interface to communicate with mobile handsets in a wireless mode. Traffic Diversion Station (TDS) uses both ad hoc technologies and cellular network technologies with one A-interface and one C-interface. In a TDS, a C-interface is used for communicating with a BS or a MS with only C-interface. It operates around 2.4 GHz in the unlicensed ISM (Industry, Science and Medical) band, and consumes out-of-band bandwidth. Mobile Handsets in the above mobile systems should be designed with more flexibility. As such there are three types of MS’s in the above mobile network: MS’s with only one C-interface or only one A-interface or both.

### 2.4 MDA-based Service Discovery for Future generation Mobile Systems

In subsection 1 and 2 of this section, some representative service discovery techniques and MDA have been reviewed. The combination of these two techniques would be very powerful to provide a more precise service description and service discovery while taking advantage of the benefits of MDA for further service composition. MDA provides models during the software development stage. These models, after transformed automatically to XMI (XML-based Model Interchange) description, can be utilized directly for service description and discovery purpose. Figure 2 illustrates this procedure. XSD, short for XML-based Service Description, is the language used to specify services in MBSD.

![Figure 2: XSD Generation Using MDA Models](image)

Newly discovered services will need to be integrated seamlessly with their existing services to provide disruptive and consistent services to user’s satisfaction. MDA’s platform- and language-neutral features fit into this scenario very well. Several works have addressed application of MDA for service creation or composition [8], though the majorities are for wired networks. However, to the best of our knowledge, no work has proposed the integration of MDA and service discovery as this paper presents. A service discovery mechanism that coincides with the service development methodology, i.e., OMG MDA, makes great impact on the future practical success of future generation mobile systems.

For example, in Figure 1, MH4 could make use of the services Y and Z from MH2 to create a new service C, which can be made available to other MHs. It becomes possible only when services Y, Z and C were developed using same methodology (MDA in this article’s case), and when a service discovery mechanism supports the exposure of development information of services Y and Z (MDA models in this case) to the service requester.

### 3. MBSD SYSTEM ARCHITECTURE

MBSD adopts a distributed architecture, namely, there is not a centralized service broke and it allows each mobile device to act as a server providing service and a client requesting services from other service providers at the same time. Figure 3 illustrates the MBSD system architecture and protocol stack.

#### 3.1 Service Description

Services first of all need to be described in terms of its main functions and interface etc. before getting involved into service discovery procedure. Existing service discovery protocols have adopted various service description languages. For example, in Jini, services are represented by a Java interface, including not only methods that clients will invoke to use the services but also descriptive attributes [3]. In SLP [3], a service is defined by a service type, address, and a set of attribute-value pairs. IBM DEAPspace [6] also utilizes this attribute-value pair style for service description, amongst many others. Due to its cross-platform feature, XML has attracted increasing attention for
service description as is the case in UPnP, WSDL, and Konark etc. XML provides richer mechanisms to express service description, therefore leading to more powerful matching for service discovery.

MBSD also adopts XML as its service description and names as XSD (XML-based Service Discovery). The following XML file exemplifies the essence of the XSD:

```
<MobileService>
  <Properties>
    <Property> ... </Property>
  </Properties>

  <AvailableFunctions>
    <Function> ... </Function>
  </AvailableFunctions>

  <Price>
  </Price>

  <XMI_Link> <-- providing a link to XMI document -->
</MobileService>
```

The basic service information is typically inputted or set by the service owner whereas the rest are generated from service XMI. There is no section in XSD for service advertisement. XMI link is empty if the service was not developed using MDA methodology. Downloadable element defines if the service provider would allow the service objects to be downloaded to the requesting client.

### 3.2 Service Registry

Service registry appears as an XML document maintaining multiple entries each describing one service provided by the service provider. Each entry contains only the basic functional description of the service, i.e., the basic service information part of the XSD document, along with a link to the full XSD of the service. The location of the registry XML file is informed of to the SDA and the SDA uses normal XML parser to retrieve elements from the XML file and carries out matching. Within a service XSD, there are links pointing to the service XMI document and service objects. Service XMI document is to be used by MDA tools to regenerate service models. The location of service objects is to be used by the SDA to deliver services.

### 3.3 Service Discovery

Each mobile device in the MBSD community has a MBSD SDA (Service Discovery Agent) that is central to the service discovery mechanism. SDA discovers required services by getting access to its local service registry. MBSD SDA does not do service advertisement as most service discovery techniques for wireless networks such as DEAPspace and Konark do. Doing so could reduce the overhead caused by service discovery mechanisms. If a SDA could not find the requested service it simply does nothing.

A mobile application interacts with the MBSD SDA (typically via local communication) to make new service component request, and then based on this request, a SREQ (Service REQUEST) message is generated and is transmitted via SOAP messaging layer to potential service providers through IP. SOAP messages, carried by XML files, are transmitted over HTTP. As illustrated in Figure 3 (b), service discovery messages can be physically transmitted via various air interfaces, such as cellular or ad hoc network interface (e.g., IEEE 802.11) depending on the hardware and software features of a mobile device. Multiple SREP (Service discovery REPly) messages might come back to the service requester bringing back more than one candidate services. In this case, Service Selection Engine (refer to Figure 3 (a)) selects the
most proper one based on each service’s reputation, price, QoS (Quality of Service) features, etc. The procedure of service discovery in MBSD is demonstrated in Figure 4, along with service delivery.

**Figure 4: Service Discovery & Delivery Workflow**

MBSD adopts a pull mechanism for service discovery. When a mobile application cannot find a component service in its local service registry, the service discovery process is triggered. Upon receiving a new service request, the MBSD client sends out a SREQ message. Depending on its knowledge of potential service providers, the application (or the use of the mobile device in most cases) can adopt one of the following three approaches for efficient service discovery. If a mobile device knows already where to get this cost-effective and quality service then unicast is utilized, i.e., the SREQ is sent directly to this service provider. On the other hand, if a mobile device knows nothing about the whereabouts of the requested service component, SREQ is broadcast. In most cases, mobile device knows a list of service providers (e.g., via its business alliance and the user’s friends list) providing the requested service components and would like to get a quote from all these SPs and then to select a most proper one. In this case, multicast is utilized. As discussed in service description section, associated with each service component are some parameters describing the features of the service component in terms of its price, reputation, QoS issues.

SREQ message and SREP message take the following format respectively:

- `<SREQ_id, source, destination_list, service_classification, keywords, max_price, service_description, options>`
- `<SREP_id, source, destination, service_name, service_id, service_classification, keywords, price, downloadable_flag, XMI_link, options>`

Maximum price is included in SREQ if the client knows the targeted SP is trustworthy. This parameter is used as one of the criteria for service matching. Options in SREQ/SREP can be QoS parameters (such as delay, media fidelity, etc.), or TTL (Time To Live) indicating the available period of a service.

### 3.4 Service Delivery

In MBSD, service delivery is carried out by a HTTP server locating at each service provider. The generic HTTP can hide the heterogeneity of different hardware and software platforms utilized by different service providers as such achieving cross-platform service discovery. SOAP (Simple Object Access Protocol) over HTTP is utilized for service discovery and delivery message passing. SOAP’s XML nature coincides with the format of service description – XML.

After the service component arrives at the client side, the latter can invoke its methods based on the service description.

### 4. MBSD Implementation & Validation

The prototype of MBSD service discovery mechanism has been implemented as part of the Panda pervasive system infrastructure in University of Essex. The experiments were carried out on two versions of Java platform, i.e., J2SE and J2ME by using two iPAQ HX2750 (OS: Microsoft Pocket PC 2003, JVM: IBM J9) and an IBM laptop X31 (PM-1.5GHz, 256M, OS: Redhat Linux 9, J2SE JVM 1.4.1), all equipped with an IEEE 802.11b wireless network adaptor. An ad hoc network is dynamically formed. Simulation on large-scale wireless mobile networks as depicted in Figure 1 was also carried out using network simulator ns-2. However the results are not presented here due to space limit.

**Figure 5: MBSD Prototype Implementation**

Figure 5 shows two graphical user interfaces (GUI) at both server side (a) and client side (b). Server GUI is used for managing the services made available to other mobile devices. For example, you can update a service by changing its name, classification (i.e., the branch on mobile service tree), keywords, etc. A unique service ID is generated automatically based on the service name, classification and the location of the mobile service. As illustrated in Figure 5 (a), you can also provide a link to the corresponding XMI document of the service and the code-base of the objects implementing the mobile services. Client GUI (Figure 5 (b)) illustrates which kinds of information are needed for service discovery. A set of APIs has been implemented to automating the service discovery process.

Suppose a user John is implementing a pervasive service that will need to play a movie clip on a mobile device that requests a MPEG4 decoder. When he finds out that he does not have the
decoder himself, he decides to use MBSD to discover the decoder because he would need to integrate the decoder into his pervasive service programmatically rather than just plug-and-play and MBSD allows him to do so. After paying a reasonable amount of fee he successfully got this service (including its XMI document and the associated implementing object classes). He imports the XMI document into his MDA tool (e.g., XMF from Xactium Ltd. [7] in our case) and regenerates the models of this decoder service as shown in Figure 6. Though John might not be aware that the service description of the decoder service was actually also generated based on these models, he knows that MBSD has helped him find precisely the service he needs.

![Figure 6: Models of the Decoder Service](image)

Please contact the authors regarding the exact content of the XMI and XSD documents of this Decoder service.

5. SUMMARY AND CONCLUSIONS

This paper proposes a model-based service discovery mechanism, MBSD, for future generation mobile systems and has provided its implementation. MBSD’s service description language XSD is not manually written by service providers in a separate phase from service development. It is generated automatically from the model information of a service that is developed based on MDA approach. So when a service has been MDA-developed, the corresponding service description is created at the same time. This integrated approach maximizes the expressiveness of the MBSD’s service description language, XSD, at no extra cost. And most importantly, this approach guarantees the semantic transition of the service to its description, whereas it is impossible in any other service description approach in current service discovery protocols.

MBSD service discovery takes into consideration the whole software development procedure, both before service discovery (i.e., the description of services) and after service discovery (i.e., service models regeneration), and as such makes positive impact on the development of software services for future generation mobile systems. The experimental validation attempts have revealed and provided evidence that the proposed methodology can be used efficiently for the discovery and development of mobile services.

References

[7] [http://www.xactium.com](http://www.xactium.com)