

## Adaptable Service Component Interface Framework in Pervasive Computing

<sup>1</sup>Vasanthi, R. and <sup>2</sup>R.S.D. Wahidabanu

<sup>1</sup>Department of CSE, Tagore Institute of Engineering and Technology, Salem, India

<sup>2</sup>Government College of Engineering, Salem, India

---

**Abstract: Problem statement:** Pervasive computing refers to visionary new ways of applying Information and Communication Technologies (ICT) to our daily lives. Interoperability among pervasive networked systems, in particular accounting for the heterogeneity of protocols from the application down to the middleware layer, which is mandatory for today's and even more for tomorrow's open and highly heterogeneous networks. **Approach:** The high demand of mobile and wireless network for Ubiquitous computing, motivate to present an Adaptable Service Component Interface Framework (ASCIF) to efficiently build pervasive computing environment suiting to recent needs of heterogeneity networks (Desktop, Web and Mobile applications). **Results:** In this Adaptable service component interface framework in pervasive computing gives, how a domain specific language for service composition can be designed to capture the common design patterns for service composition. **Conclusion:** Adaptable Service Component Interface Framework for constructing pervasive computing environment to fulfils the interoperability and manageability requirements of Web services based applications. The interactive service components are developed as application programming interface entities to bind with different heterogeneity environment.

**Key words:** Middleware framework, pervasive computing, service components

---

### INTRODUCTION

Pervasive computing (Saha, 2005) is “omni-computing”. It is “all pervasive” by combining open standards-based applications with everyday activities. computing is a rapidly developing area of Information and Communications Technology (ICT). The term refers to the increasing integration of ICT into people's lives and environments, made possible by the growing availability of microprocessors with inbuilt communications facilities. Pervasive computing has many potential applications, from health and home care to environmental monitoring and intelligent transport systems. Pervasive computing systems (PCS) and services may lead to a greater degree of user knowledge of, or control over, the surrounding environment, whether at home, or in an office or car. They may also show a form of ‘intelligence’.

Middleware means any piece of software that glues together various other pieces of software can be labeled as middleware (Fjellheim *et al.*, 2005). The two most common functions handled by middleware solutions are messaging and data access services. A typical usage scenario is one where a Graphical User

Interface (GUI) component needs to access a remote database. Usually the GUI part has to be independent of the actual database implementation and a middleware component or a set of middleware components provide that functionality to the GUI. Thus middleware provides a service layer in the software architecture that separate the details of implementation from users of middleware in Fig. 1. The typical users of middleware are application developers who build new applications to be deployed in the target environment.

Interoperability among systems is fetching more demanding in the Web era since the enlarged point of connectivity, the information space is huge and dynamic, the information formats and semantics are growing more diverse, organization are independent and there is a require for incorporation to be straightforward, rapid, protected and flexible to transforms. Communications among heterogeneous and independent applications want interoperability at three layers such as interaction, content and business process layer. Cooperating applications require agreeing on their combined business process, the content of the replaced documents as well as the message protocol to trade messages.

---

**Corresponding Author:** Vasanthi, R., Department of CSE, Tagore Institute of Engineering and Technology, Salem, India

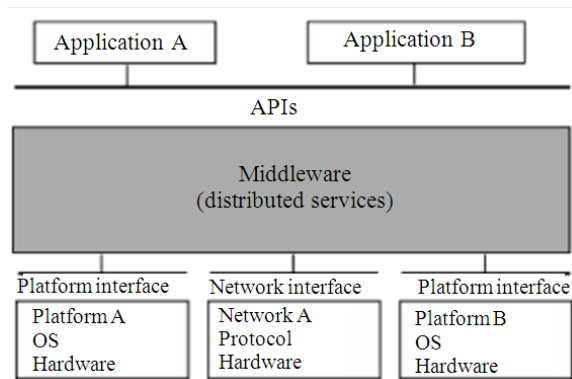


Fig. 1: Middleware

Web services suggest a physically powerful base for software interoperability via the core open standards of Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), Web System Description Language (WSDL) and Universal Description, Discovery and Integration (UDDI).

Moving Web services away from straightforward end-to-end applications to Internet-scale application-to-application interoperation needs a pattern move from object-based architecture to an insecurely-coupled, information-focused and service-oriented architecture. This study presents an Adaptable Service Component Interface Framework (ASCIF) to efficiently create pervasive computing environment suiting to recent needs of heterogeneity networks. It provides perceptive of Web services interoperability confronts that arbitrates Web services communications to increase interoperability and reliability. ASCIF includes value and openly addresses few of the mainly serious issues surrounding the venture-grade employment of Web services.

**Related works:** The services incurred from the middleware components recently needs to adapt with web application, mobile application and desktop as well. Most of the recent literary works built web services suiting the features of middleware component requirement for pervasive computing (Mokhtar *et al.*, 2007). Most becoming efforts recognize the require to expand the present methodological infrastructure to foster wider adoption of Web services (Maheshwari *et al.*, 2004). Gupta *et al.* (2002), explore enhancing Web services by presenting Message Oriented Middleware (MOM) in the architecture.

Achieving software interoperability (Gupta *et al.*, 2002) at various layers has been there for years and stay remained a main research focus. Before Web services entered, there have already been number of

prior technologies like EDI (Electronic Data Interchange), Common Object Request Broker Architecture (CORBA), Distributed Component Object Model (DCOM) and Java Remote Method Invocation (RMI) with limited success. Neither of the previous methods (Dressler *et al.*, 2009) will gain wide industry helps and access because of their toughness (e.g., CORBA, DCOM), cost factors (EDI), or platform denotes (e.g., RMI). Nevertheless, these attempts generated tightly coupled systems that are not appropriate for Business to Business (B2B) incorporation (Capra *et al.*, 2003). Web services suggest overcoming problems by encouraging standard-based interoperability and movable coupling. Web services are software components (Davies *et al.*, 2004) that depict several computational or data offering potential in a standard format and available using standard Internet protocols (Desai *et al.*, 2005).

Web services interactions are based on replacing standards-based XML documents (i.e., message-based interoperability). Interoperability is one of the major features that Web services assure (Hadim and Mohamed, 2006). Along with its open standards, Web services authorize flawless communication between applications in a proposal and language neutral way (Zender *et al.*, 2008). There is a obvious want for novel wide-area messaging infrastructure to improve service delivery in web-centric environment. Web Services Message Bus (WSMB) is a primary pace towards achieving this (Nakazawa *et al.*, 2006). Its intend was firstly inspired from ideas introduced by the Web services Invocation Framework (WSIF) Grimm *et al.*, 2004) but it appends more advanced QoS features as well as support for message-oriented interactions (Ranganathan *et al.*, 2005) instead of RPC-based invocations encouraged by WSIF.

Web services can simply adjust existing MOM technologies to attain higher QoS properties, given that they try to address interoperability on a scale at which conventional technologies have failed. Adaptable Service Component Interface Framework intends to expand and leverage the principles of messaging systems as well as the promising Web services standards to carry more interoperable and manageable Web services communications using a decentralized architecture.

## MATERIALS AND METHODS

Adaptable middleware framework for desktop-web-mobile interactive services: For pervasive computing, Web services acts as the grand mediator. Pervasive computing architectures diverge deeply in their diffident compatibility for offered systems.

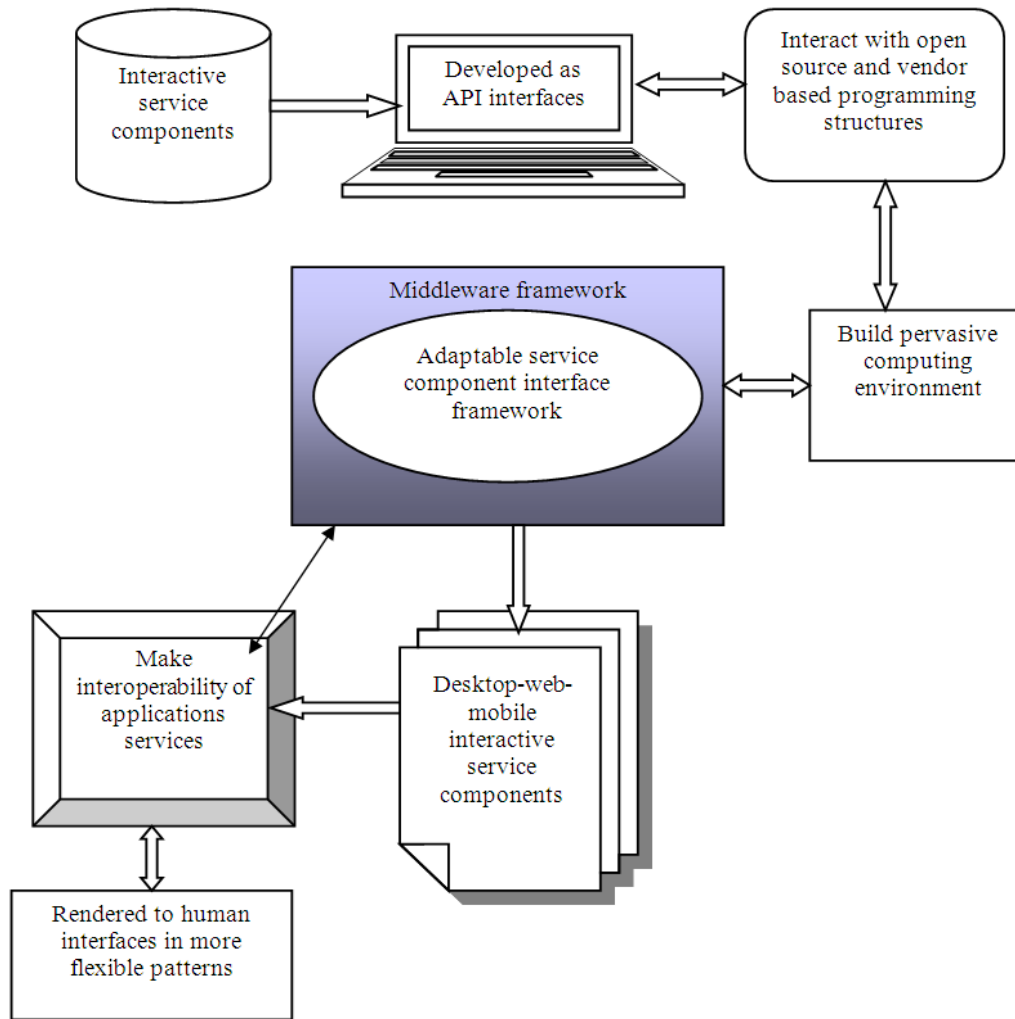


Fig. 2: Adaptable Middleware Framework for Desktop-Web-Mobile Interactive Services

HTTP is assumed as the standard transport protocol for conveying information. Consequently, Most of the current Web services frameworks like Axis are almost exclusively geared to synchronous invocation over HTTP which might not be suitable for applications with high reliability requirements. As a middleware bridging between clients and Web service providers, ASCIF provides a configurable framework for reliable Web services interactions. ASCIF provides different channels to extract the registered Web services in which every web service is coupled to one or more channels.

The incoming information is reviewed on entrance via the channel to decide the destination service. Filters hurdle to the destination service if any interrupt and operate both appeal and reply information in which transforms previous format messages into latest formats. The message is next passed via a

reliability layer where it is ensured for termination, repetition and ordering and then it obtain queued for processing. ASCIF afterward send out the message to the destination Web service and the reply is passed back to the asker via the similar path.

ASCIF in Fig. 2 controls interception of messages and protocol connection to permit interactions over numerous communication protocols (e.g., HTTP, JMS, MSMQ...). The option of the suitable protocol to utilize will be chosen depends on the necessary Quality of Service and obtainable infrastructure. ASCIF permits the Web service client to prefer the beloved transport with no directly interfacing any organized middleware which transports information. ASCIF clearly uses the proper communication channel, from the method of the endpoint URL of the destination Web service. ASCIF

allows a client to evidently transmit a message to queue as it had to an http-based endpoint. ASCIF illustrates the capability to efficiently control between transports protocols by merely changing the intention endpoint from a HTTP URL, to a listener offering on JMS or Websphere MQ queue in synchronous and asynchronous manner.

Multi-protocol carry assists to conserve interoperability while leveraging offered infrastructure. For instance, a procure order summarized to message will enter using HTTP followed by ASCIF put the message in a message queue to be pickup, authorized, scheduled and processed. The requesting process will ultimately accept a message verifying or discarding the order via ASCIF mediation. This permits protecting outside users and associates from the internal implementation details while maintaining data losses to a minimum.

### RESULTS

Performance evaluation on adaptable service component interface framework, Adaptable Service Component Interface Framework is implemented by using the Java platform. The performance evaluation tests aimed at comparing the direct invocation using HTTP with channeling interactions through ASCIF. ASCIF infrastructure is depended on interception (Fig. 1). At set up it loads the arrangement and begins a group of addressees to start receiving messages. The addressees read messages from various channels and then call a sequence of configurable managers to operate together request and response messages as educated by the arrangement settings. Messages travels along a configurable channel made of a sequence of operations for processing and relaying messages. ASCIF carry two types of operations generic (like confirm a message against a schema, alter a message with XSLT, divide, route, encrypt/decrypt message...) and specific (like a tradition handler to estimate or impose a business rule). Operations can be assigned to different nodes in the infrastructure. Each node receives the messages into a personal queue and processes them through a configurable sequence of handlers, after that the message will depart the node to a definite destination or set of destinations.

The performance of ASCIF is evaluated by the following metrics:

- Number of applications
- Users load-demand variance
- Binding duration

### DISCUSSION

In this study, how a domain specific language for service composition can be designed to capture the common design patterns for service composition have explored, yet still retain a comparable performance to other systems written in mainstream languages such as Java. ASCIF run independent tests with growing number of applications and constant number of requests sent by each client to 50 requests per client.

Web services allow seamless communication between applications in a platform and language neutral way. The various numbers of applications are used in the experimentation to validate ASCIF. The performance graph which is shown in Fig. 3 illustrates Comparison result of ASCIF with WSMB based on users load demand variance, measured in terms of Kilo Bytes (KB). As number of applications increased, Users load demand variance gets decreased in both ASCIF and WSMB. As compared with WSMB, ASCIF outperforms well.

In Fig. 4 depicts the performance graph which compares ASCIF with WSMB by using the metrics users load demand variance and Binding duration. Load demand variance measured in Kilo Bytes (KB) and binding duration is measured in seconds. The binding duration is high in ASCIF when compared with WSMB.

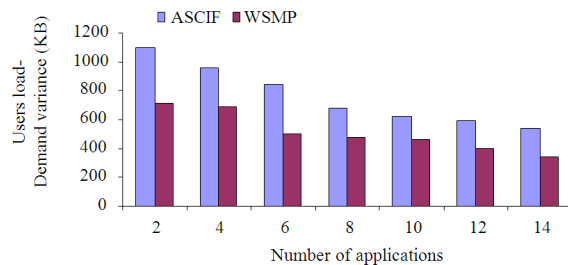


Fig. 3: Comparison result of ASCIF with WSMB based on users load demand variance

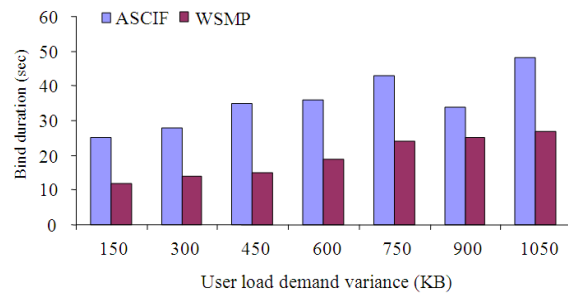


Fig. 4: Comparison result of ASCIF with WSMB based on the Binding Duration

## CONCLUSION

Adaptable Service Component Interface Framework for constructing pervasive computing environment to fulfils the interoperability and manageability requirements of Web services based applications. The interactive service components are developed as application programming interface entities to bind with different heterogeneity environment. API interfaces have the flexibility to interact with open source and vendor based programming structures in pervasive computing of web, mobile and desktop applications. In addition efficacy of our work is measured with metrics such as number of applications, users load-demand variance, binding duration, service level agreements and number of heterogeneity services. Standard benchmark data sets are taken from UCI repository (i.e., web navigation and mobile usage) to conduct the performance evaluation.

## REFERENCES

- Capra, L., W. Emmerich and C. Mascolo, 2003. CARISMA: Context-aware reflective middleware system for mobile applications. *IEEE Trans. Software Eng.*, 29: 929-945. DOI: 10.1109/TSE.2003.1237173
- Davies, N., A. Friday and O. Storz, 2004. Exploring the grid's potential for ubiquitous computing. *IEEE Pervasive Comput.*, 3: 74-75. DOI: 10.1109/MPRV.2004.1316823
- Desai, N., A.U. Mallya, A.K. Chopra and M.P. Singh, 2005. Interaction protocols as design abstractions for business processes. *IEEE Trans. Software Eng.*, 31: 1015-1027. DOI: 10.1109/TSE.2005.140
- Dressler, E., R. Zender, U. Lucke and D. Tavangarian, 2009. A multi-layer approach for cross-technology communication in a pervasive community. *Proceedings of the 6th Annual International Mobile and Ubiquitous Systems: Networking & Services, MobiQuitous*, Jul. 13-16, IEEE Xplore Press, Toronto, ON., pp: 1-6. DOI: 10.4108/ICST.MOBIQUITOUS2009.6996
- Fjellheim, T., S. Milliner and M. Dumas, 2005. Middleware support for mobile applications. *Int. J. Pervasive Comput. Commun.*, 1: 75-88. DOI: 10.1108/17427370580000114
- Grimm, R., J. Davis, E. Lemar, A. MacBeth and S. Swanson *et al.*, 2004. System support for pervasive applications. *ACM Trans. Computer Syst.*, 22: 421-486. DOI: 10.1145/1035582.1035584
- Gupta, S.S., F. Karim, Y. Wang, B. Wang and S.K.S. Gupta, 2002. Reconfigurable context-sensitive middleware for pervasive computing. *IEEE Pervasive Comput.*, 1: 33-40. DOI: 10.1109/MPRV.2002.1037720
- Hadim, S. and N. Mohamed, 2006. Middleware: Middleware challenges and approaches for wireless sensor networks. *IEEE Distrib. Syst.*, 7: 1-1. DOI: 10.1109/MDSO.2006.19
- Maheshwari, P., H. Tang and R. Liang, 2004. Enhancing web services with message-oriented middleware. *Proceedings of the IEEE International Conference on Web Services*, Jul. 6-9, IEEE Xplore Press, pp: 524-531. DOI: 10.1109/ICWS.2004.1314778
- Mokhtar, S.B., N. Georgantas and V. Issarny, 2007. COCOA: CONversation-based service COMposition in pervASive computing environments with QoS support. *J. Syst. Software*, 80: 1941-1955. DOI: DOI: 10.1016/j.jss.2007.03.002
- Nakazawa, J., H. Tokuda, W.K. Edwards and U. Ramachandran, 2006. A bridging framework for universal interoperability in pervasive systems. *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems, (ICDCS' 06)*, IEEE Xplore Press, pp: 3-3. DOI: 10.1109/ICDCS.2006.5
- Ranganathan, A., S. Chetan, J. Al-Muhtadi, R.H. Campbell and M.D. Mickunas, 2005. Olympus: A high-level programming model for pervasive computing environments. *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications*, Mar. 8-12, IEEE Xplore Press, Kauai Island, Hawaii, pp: 7-16. DOI: 10.1109/PERCOM.2005.26
- Saha, D., 2005. Pervasive computing: A vision to realize. *Adv. Comput.*, 64: 195-245. DOI: 10.1016/S0065-2458(04)64005-8
- Zender, R., E. Dressler, U. Lucke and D. Tavangarian, 2008. Meta-service organization for a pervasive university. *Proceedings of the 6th IEEE International Conference on Pervasive Computing and Communications*, Mar. 17-21, IEEE Xplore Press, Hong Kong, pp: 400-405. DOI: 10.1109/PERCOM.2008.74