CHAPTER 4

INTEROPERABILITY

Interoperability refers to the concept of “two co-operating process working together without interfacing” [31]. To match the semantic context is challenge of interoperability. Interoperability is an objective addressed during design, development and enhancement of distributed system and requires middleware during run time. Vallecillo, Hernandez and Troya identified two properties of service interoperability - Comparitability and substitutability. Comparitability is the property in which two interacting components work together. Substitutability is the competence of service based on environmental related context information. This service may be substituted by another one without violating contextual service interoperability. The heterogeneity between different middleware platform and part of application running on different platform seeks interoperability. There are no tools available to support interoperability [31].

In the proposed work on interoperability, the major emphasis is done on the service discovery protocols. Each smart space of the transport layer rest of a service discovery protocol. Interoperability of the service discovery protocol is attained at the middleware. Interoperability could be of two forms - interoperability across middleware platforms and interoperability across the protocols that acts as a substratum in each platform. In the proposed work we have not considered the interoperability across different middleware platforms.
4.1 Interoperability on Smart Spaces

Smart spaces are active networks composed of mobile nodes, actuators and other mobile devices. Interoperability is a phenomenon by means of which all heterogeneous smart spaces are brought together. Each of the smart space designed in our work are heterogeneous spaces. They differ in the behavior of the protocol in which they rest. Each of the smart space has a service discovery protocol as a substratum. And no two service discovery protocols have got the same principle of working. Hence to bring these different heterogeneous spaces together - interoperability is an essential feature. Interoperability of protocols is done seamlessly without any bridges or bindings. Seamlessness is added to the code of the routing protocol and as the protocol gets downloaded automatically they interoperate in the middleware inside the discovery module. The only disadvantage is heavy overload due to code interoperability but this is reduced due to the sensor enabled pervasive services, provided by architecture. As asynchronous seamless delivery of service is inculcated in our architecture, the overhead is compensated and reduced. Thereby seamless interoperability proves a better result in pervasive adhoc environment where sensors are deployed.

The proposed middleware provides interoperability among several service discovery protocols. Mobile environment supports light weighted middleware. But addition of more and more components increases the size of the middleware and makes it heavy. The automatic discovery service test enables the selection of a particular service discovery and routing protocol depending on the environment and context of the user.
Nodes in Fig 4.1, interacts with computation entity seamlessly through service discovery protocols. The concept of interoperability in service discovery framework is publish-subscriber model. Functional elements of the proposed architecture proved to be efficient than the architecture model of [32] which involved handlers. The proposed model consists of entity named generator, parser and composer. Generator generates the occurrences of services which are seamlessly passed in to the parser and the parser finds the exact match from the available service both syntactically and semantically. The application obtains request from the user by analyzing the profile of the user and seamlessly adds up context information through the monitor component and then finds the match with the help of parser and routes it to the user through the composer. In the proposed work the multidirectional propagation through all the cluster heads are made possible. The cluster that exactly matches the service is selected. The composer hides the transparency of interoperability with the outside world thereby making it seamless.
Fig. 4.1 Interoperability Across Nodes of Smart Spaces.

The selection of the protocols is done by automatic cycling test. Thus service discovery interoperability occurs in the service discovery component framework at the middleware. Once the service selection is established then communication happens spontaneously between the peers and the services are retrieved to the user. The routing interoperability is attained through code mobility. Interoperable functionality is also downloaded along with the routing code when required.

4.2 Interoperability on Cross layering Architecture

By deploying a cross layer at the middleware the interoperability between the service discovery protocol and routing protocol is attained. The Fig. 4.2 depicts the interoperability of SLP and AODV, and other combination of PDP and DSDV.
Multiple protocols are made to interoperate across the platform at a particular instant of time. Protocols are made to interoperate irrespective of the language, features and other uniqueness. The interoperable architecture is not confined to one language like java based, or one form like object based. Any protocols could be made to interoperate at any time. Interoperability between service discovery and routing protocol is through the cross layer approach at the middleware. If two protocols offer similar service then the best matching service is offered to the user with the help of sensor nodes. The heavy overload of profile is only at the header node of the peer to peer cluster. The header selects the service from within its cluster or from any other cluster and offers it seamlessly to the user. But the overhead is minimized due to seamless interoperability rather than query response type of communication. The query response type of communication incurs heavy overhead in the existing methods.

This chapter deals with the next important feature of the middleware – “The Interoperability”. Interoperability of the service discovery protocol first occurs at the middleware and by the cross-
layer approach interoperability of the routing protocol is also taken from routing layer.

4.3 Conclusion

The interoperability of the protocols brings different heterogeneous smart spaces together. The functionality of interoperability and cross layering increases the efficiency of the middleware and thereby takes minimum time for search and retrieval of data. The next chapter deals with cross-layer approach.