How to Put an Elephant in a Refrigerator: Architectural Concerns of an ESB for Lightweight Environments

Hasini V. Abeywickrama[#1], Chamila K. Adhikarinayake[#2], Gimantha M. Bandara[#3], Samitha R. Senarathna [#4], Shahani M. Weerawarana [#5]

[#]*Department of Computer Science and Engineering, University of Moratuwa Moratuwa, Sri Lanka*

[1]hasinia.09@cse.mrt.ac.lk
[2]chamila.09@cse.mrt.ac.lk
[3]gimantha.09@cse.mrt.ac.lk
[4]samitha.09@cse.mrt.ac.lk
[5]shahani.w@gmail.com

Keywords— **Enterprise Service Bus, Internet of Things, Event driven model**

INTRODUCTION

With the introduction of IPV6, the address space has increased. Thus in the future, every electronic device can have a unique address. These devices would generate data every day. We can use this data to generate information and create intelligent systems. The concept related to this scenario is known as the Internet of Things (IoT). Interconnecting a multitude of these devices and sharing data and services will eventually become a significant challenge. Enterprise Service Buses (ESB) which are used to integrate enterprise applications can be used to address this challenge by interconnecting every device in a more scalable manner. Even though existing ESBs are suitable for enterprise application integration, they may not be suitable for lightweight hardware environments.

Internet of Things (IoT) is an emerging technology in which uniquely identifiable objects interact with each other over a network. IoT technology can be used in industries to automate the working process and in homes to make them smart homes. In IoT concept objects interact with each other through various technologies. We can see some of the requirements for the IoT implementations can be satisfied with an ESB. An ESB can be used to maintain connections within a subsystem as well as interconnect two or more subsystems with each other more efficiently.

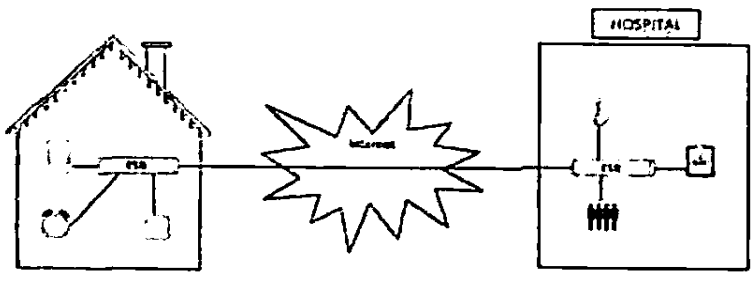

Fig. 1 Using ESBs to interconnect devices

There currently exist many good and popular ESBs, competing with each other. Each tries to provide something unique in order to increase performance. Many performance comparison tests were carried out to find which ESB is the best in performance. But these solutions are suitable for enterprise environment. They cannot be run on a low resource environment handling large number of concurrent connections due to their higher resource consumption.

To address these issues, a new ESB needed to be implemented. This ESB should be designed to run on a light weight environment and handle high levels of concurrent connections.

Some of the architectural concerns for such an ESB for a lightweight environment are listed below:

- Changing the architectural concurrency model from thread based architecture to event driven architecture model.
- Changing the programming language by using a language which runs on a lightweight environment and provides an event driven programming facility.

High levels of concurrent connections can be achieved by introducing a design using event-driven architecture instead of thread-based architecture. Event driven design is much suited for low resource environment.

The programming language can be a deciding factor when it comes to developing an ESB to suit a light weight environment. Programming languages such as Java can consume more resources than other languages. Dynamic programming languages like JavaScript can be used as an alternative language. Node.js is one of the emerging frameworks which use JavaScript for programming. Thus a purely event driven ESB which is based on Node.js would be a good solution for the issue. The pure event driven nature makes the ESB low resource consuming while the use of JavaScript increases the ability of running on a light weight environment.