An Effective Framework for Integrating Web Services with Embedded Systems

B. V. S. Sai Chaitanya*

Department of Electronics and Computer Science Engineering, KL University, Vaddeswaram Guntur – 522502, Andhra Pradesh, India; bvssaichaitanya@gmail.com

Abstract

Background/Objectives: An efficient architecture of a miniaturized WEB services server which can be made to be resident on a micro controller based system so that embedded systems can be made to provide the support of WEB services that are related to sensing and measuring various environmental factors. Methods/Statistical analysis: The proposed architecture deals with two parts which include client side and second includes embedded server side part. In client side part a java based UDDI server is implemented on the PC. On server side two threads are created as individual functions which are responsible for receiving the client request and for transmitting the result. Findings: Developing and deploying WEB services requires huge amount of resources which are not supported by any embedded systems. An RTOS based task oriented services and thread oriented communication interfaces reduces the overhead and leads to implementation of light weight WEB services server. Application/Improvement: In future the provision of many threads each meant for a client has to be made the architecture must support several pairs of threads for communicating with several clients.

Keywords: Client and Server Communication, Light Weight WEB Services Server, RTOS Based Services, Thread Oriented Communication, WEB Services within Embedded Systems

1. Introduction

Embedded systems are seldom used alone. Systems that communicate between each other are common in real time ubiquitous applications. If the embedded systems could directly work together with other computing devices, they would add value to each other, and enable new consumer application. Present requirements of cyber-physical systems are high for implementing them on single, non-networked units. The usage service oriented architecture is one of the solutions to achieve interoperability and possible future scaling of the system. A limited subset of XML WEB service protocols can be implemented. WEB services put forward a new architecture model for applications and enact capabilities that are available to the other applications via protocols, industry standard network and application interfaces. Surprisingly they found that limited XML WEB service implementation introduces minimal overhead. Using XML WEB services as the communication layer for embedded systems facilitates standardization of communication and helps to increase interoperability between different types of communicating systems. According to Metcalfe's law, the value of the system is larger than the sum of its components. An application can use the knack of a WEB service by simply implore it across a network without unification. The services are registered into world known registries and the whereabouts of the services can be known by enquiring the registries. Therefore the services can be conveniently be accessed by using a standard protocol. WEB services are described through a specific language (WSL). The details of the WEB services can be obtained by enquiring UDDI registry, the communication between the client and the services server which is implemented through another standard language called SOAP. Specialized embedded systems are being developed for processing environmental factors such as temperature, pressure, humidity, speed, flow, level etc., which are tried
and tested and the quality of such systems are proved everywhere. However the embedded systems are limited in resources and therefore miniaturized WEB services systems must be designed and implemented. The below Figure 1 illustrate the WEB service architecture.

Figure 1. WEB service architecture.

There are two approaches for architecture of WEB service. The pivotal aspect is to explore the restricted area of it and another contour is WEB service protocol stack. Typical architecture comprises of three distinct parts which grasp contributor, petitioner and registry concerning the particular service. Another approach is to find the rising WEB service protocol stack. The Figure 2 describes the WEB service protocol stack.

Figure 2. WEB service protocol stack.

2. WEB Service Technologies

2.1 UDDI

UDDI currently reorients the discovery layer inside the WEB service protocol stack. UDDI includes a technical stipulation for building a distributed directory of business and WEB services and the UDDI business registry is absolutely operational execution of the UDDI stipulation. Data is accumulating inside a specific XML form. The UDDI stipulation involves API details for discovering existing information and post new information. Data encapsulate inside UDDI is ordered into three fundamental classes they are white pages, yellow pages and green pages.

2.1.1 XML

XML has set up onto the computing area in contemporary years. It has acquired expeditious approval because it validates diverse computing system to share data most easily, regardless of operating system or programming language. There are numerous number of XML tools, containing parser and editors that are accessible for each operating system and programming language like java, C, C#, C++ etc.

2.1.1.1 XML-RPC

XML-RPC is a straightforward protocol that uses XML messages to perform RPCs. XML encodes the requests and communicated suing HTTP POST. The body of HTTP response is embodied with response of XML. As it is known that XML-RPC is independent of the working platform and easy communication of various applications is facilitated.

2.1.2 SOAP

The inter PC data transfer is enabled using SOAP which is basically an XML protocol. The remote procedure calls are delivered through HTTP using SOAP. It can be used in messaging systems and transported using transport protocols. The remote methods are invoked and remote services are connected to corresponding cliental applications using SOAP. Language translation feature is enabled using appropriate method with the help of SOAP. The figure 3 describes the WEB service environment
Figure 3. Environment.

2.1.3 WSDL

WSDL is a specification which describes four crucial data, namely interface information, data type information, binding information and addresses information. In a nutshell, it defines the description of WEB services in XML. WDSL is acts ladder or a bridge between the client and the service provider, similar to Java. The contrast between these two is that WDSL is language and platform free. It is used basically to describe SOAP services.

Figure 4. WEB service discovery, bind and utilize.

The client specified WEB service is located by calling the public functions. The integration of applications with services leads to process automation using WDSL – aware tools. WSDL provides common service description language and also platform to automatic service integration. Therefore it is the pivot of the architecture of WEB services.

3. Communicating using TCP/IP and UDP Built within SOAP

UDP and TCP are standard, well-supported protocols for computers that need to send and receive information with in local network or on the internet. TCP or UDP is used to send data related to application protocols. For illustration, IP address request by PC is sent to DNS server which is placed in a UDP datagram. The Web page request is accepted and the corresponding webpage is transported in TCP segments. UDP and TCP are used for data transfer. UDP in general is not a complex protocol for implementation. It does not built in support for determination of message acknowledgement, message flow order and message flow control. The programming for protocols is similar for modules with UDP or TCP compatibility. UDP, TCP, or other Internet protocols must use IP addresses for Identification of the sender and receiver of the communications (with the exception that a UDP datagram doesn't have to specify a source address). The device firmware may specify the values (a net mask value, the IP address of a gateway, or router, and the IP address of a domain-name server) or the device may request the values from a DHCP server. Hypertext Transfer Protocol (HTTP) is standard application protocol for serving the Web page request by PC.

3.1 Implementing WEB Services within an Embedded Board (RABBIT)

The network interface supported by Dynamic C fetches an IP address and related values for Rabbit board. An application can provide the values or obtain values from a DHCP server. The program code that specifies the values or how to obtain them can be in the main application or in a macro that the main program calls. In order to keep main program free of system-specific values, macros are used. Macros facilitate the usage of same configuration in multiple programs, and enables changing a configuration by just specifying a different macro.

3.2 Problem Definition

Implementing WEB services server with in an embedded systems leads to extensive automation and wide usage. Implementation of WEB services requires huge amount of storage area processing power and porting of the technologies which are required for interacting with internet world. Embedded systems are limited in resources and therefore throw a challenge to implement light-weight embedded WEB services that can scatter for the user requirements. The main problem thus is designing WEB services and implementing the same such
that they operate within the available resources situated in embedded systems. The WEB services implementation must be miniaturized requiring minimal resources eliminating overheads.

4. Challenges for Implementing WEB Services within Embedded Systems

Embedded systems once in a while have enough memory and processing power to run a web services. Then again, current web services executions don’t adequately apply to the embedded processing. The additional overhead is brought on by the additional size of the messages that must be transferred, when compared to the actual size of the data carried within such messages. This includes abstractions on the underlying network protocols and serializations of the data. The XML web services protocols makes an entire new arrangement of design trade-offs and issues for creating web services applications for embedded systems.

In that case many embedded systems behave like server engaged in collecting the process data and process the same before the data transmitted to a remote location. Some of the embedded systems can be designed to collect the environment data and transfer the same to those who require such data. Therefore there is a need to design a service oriented architecture that is light weight and still be above to support the entire platform required for implementing the WEB services.

5. A New Architecture for Implementing WEB Services within Embedded Systems

- **Client side**
  This part is to be implemented on a PC. A java based UDDI server is implemented on the PC. The four WEB services shown in the architecture diagram are to be developed in terms of WSDL which are to be posted on to a UDDI server. The interface required to post the server is provided within the User application on the client side.

  The client application has in it two threads which are meant for transmitting and receiving the message. The request for the service is transmitted using SOAP message format to the embedded web server. The reply sent by the WEB server is received through another socket. The received message is enclosed and used by the application for further processing.

- **Server side**
  On the server side all the services are coded as individual tasks which are created under operating systems. Two threads are created on the server side as individual functions which are responsible for receiving the client request and for transmitting the results. The thread control task controls several client requests received by the sever on concurrent basis. All the tasks are scheduled under an RTOS and the tasks are invoked as per the requirements. By default all the tasks are in blocked state and the tasks are invoked as when a request is received related to the service.

![Figure 5. Embedded web services architecture.](image-url)
6. Conclusion

Implementation of WEB services in terms of an embedded system is absolutely required as many environmental factors which are frequently sensed by designated embedded systems are frequently required by different application. Developing and deploying WEB services requires huge amount of resources which are not supported by any embedded systems. An RTOS based task oriented services and thread oriented communication interfaces reduces the overhead and leads to implementation of light weight WEB services server.

7. References

1. Birkmose MS, Kristensen LH. Producing efficient web services for distributed embedded systems. 2004 Jun.
17. Kyusakoy R. Efficient WEB services for end-to-end interoperability of embedded systems. Lulea University of Technology. 2014.