Computers in Industry 71 (2015) 58–76

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

A middleware for industry

Djamel F.H. Sadok *, Lucas L. Gomes, Markus Eisenhauer, Judith Kelner

Networking and Telecommunications Research Group, Universidade Federal de Pernambuco, Brazil

ARTICLE INFO

Article history:
Received 19 December 2013
Received in revised form 7 November 2014
Accepted 13 March 2015
Available online 13 April 2015

Keywords:
Computer networks
Middleware
Industrial networks
Wireless sensor networks

ABSTRACT

This paper describes an innovative distributed framework for monitoring and control of large-scale systems by integrating heterogeneous smart objects, the world of physical devices, sensors and actuators, legacy devices and sub-systems, cooperating to support holistic management [1]. Its featured Service Oriented Architecture (SOA) exposes objects' capabilities by means of web services, thus supporting syntactic and semantic interoperability among different technologies, including SCADA systems [23]. Wireless Sensor and Actuator Network (WSAN) devices and legacy subsystems cooperate while orchestrated by a manager in charge of enforcing a distributed logic. Particularly crafted for industrial networks are new middleware services such as dynamic spectrum management, distributed control logic, object virtualization, WSANs gateways, a SCADA gateway service, and data fusion transport capability. In addition, new application oriented objects such as shop floor, manufacturing line, welding station, robots, and cells have been introduced in the middleware. The combination of such objects and previous modules offers a new and flexible industry oriented middleware. A second contribution is in the form of traffic analysis conducted at the floor level. It shows the dominance of some end systems such as PLCs, the presence well behaved almost constant traffic made up of small packets.

1. Introduction

The need for high quality manufactured goods and increased productivity are chief drivers for the advance of manufacturing systems. The emergence of IT distributed systems paved the way for distributed control in manufacturing lines using intelligent sensors and actuators. Information technology trends have a direct impact on manufacturing control. For example, recent management paradigms such as self-configuration and self-organization are also entering the industrial networks arena [2].

Despite the ongoing advances in middleware design, there is, however, a relative lack of IT standards for the manufacturing domain. Before we examine the current industrial middleware state of the art, we present recent contributions toward middleware design for traditional data networks. We show the gap separating the two worlds: a fast IT domain and a manufacturing track, which is slower to adapt. We then point to current efforts for bridging the gap, especially the work within the BEMO-COFRA project [1].

We briefly introduce middleware evolution in this section followed by industry grade middleware in Section 2. Sections 3 and 4 detail our proposed middleware and present the results of a simple packet level traffic analysis made at a factory floor, respectively. The last section concludes our work and points to some future directions.

1.1. Middleware evolution

Current middleware evolved from the simple remote procedure call (RPC) to message passing middleware architectures required to ensure reliable delivery, correct ordering and non-replication of messages. Next, and for more than two decades, object oriented middleware architectures have been put forward. These offer a more powerful programming environment, are more complete, and have built-in service discovery functionality. Examples include the Object Management Group (OMG's) CORBA [25] and Microsoft's Distributed Component Object Model (DCOM) [26]. More recently, the publish/subscribe paradigm was embraced by Internet applications [27]. Furthermore, applications handling considerable amounts of data stand to benefit today from the Map and Reduce concept, provided that the data may be operated on in parallel [28].

The Service Oriented Architecture SOA is a collection of underlying Web technologies, described using the Internet

* Corresponding author. Tel.: +55 81 2126 8954; fax: +55 81 2126 8955.
E-mail addresses: jamel@gprt.ufpe.br (Djamel F.H. Sadok), Lucas.gomes@gprt.ufpe.br (L.L. Gomes), Markus.Eisenhauer@fit.Fraunhaufer.de (M. Eisenhauer), jk@gprt.ufpe.br (J. Kelner).

http://dx.doi.org/10.1016/j.compind.2015.03.008

eXtensible Markup Language (XML), and responsible for the creation of a middleware capable of *composing* Web applications and services according to their requirements. This process is sometimes referred to as *orchestration*. An important additional development to SOA is that of an enterprise service bus (ESB) [3]. Simply put, the ESB translates a message to the correct message type and sends it to the right producer service (see Fig. 1). It therefore acts as a broker among a number of programming frameworks.

Despite not formally seen as a standard, a lightweight de-facto middleware known as the Representational State Transfer (REST) has received considerable momentum among Internet software developers. It uses a stateless, client server, cacheable communication protocol. This is usually HTTP. However, REST is more than a protocol; it is a lightweight architecture for the design of networked applications and avoids complex mechanisms adopted by many of the existing middleware architectures.

REST is built around the transfer of representations of resources. Any addressable concept is seen as a resource and its representation is a document that captures its current state. Hence, a resource may receive queries in different representations as well as answering them in equally different representations when needed. For example, Fig. 2 shows a dispatcher process storing a new state in the XML format whereas the WebApp is reading this in the also commonly used JavaScript Object Notation (JSON) hence there is a clear separation between the resource and its different representations. This is exactly what gives REST its strength. The client or application needs to understand the format of the resource returned.

Interoperability among objects through a common logical bus such as in the case of ESB or using a REST representation are at the heart of middleware design. The REST approach seems more

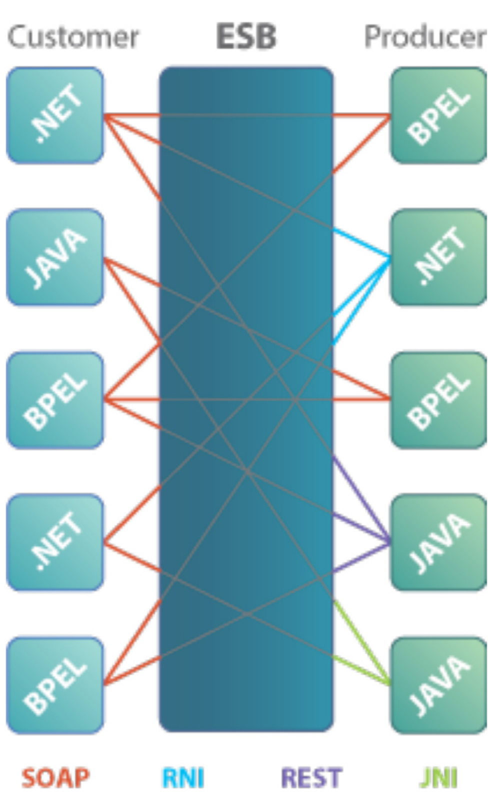

Fig. 1. Enterprise service bus as a broker among different distributed development environments.

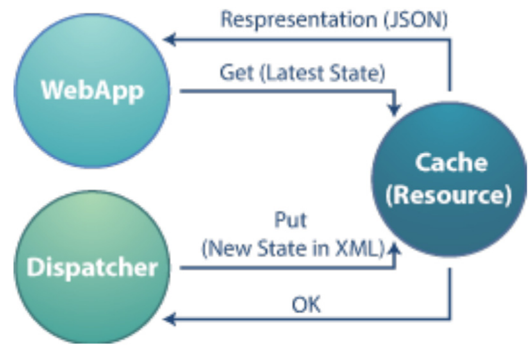

Fig. 2. REST resource representations.

attractive as it is both lightweight and enjoys wide support in the Internet.

2. Industry grade middleware

Unlike Internet services, relatively less effort was spent on the design of suitable middleware solutions, in the context of industrial environments. Industrial applications have stringent communication reliability, tolerance to interference and geometry constraints. Note that most existing works concentrate on the communication side of the problem, which, in this case, is mainly wired sensor networks. Very few tackle further important concerns such as new protocol design with real-time constraints, node addressing, integration, and support of legacy systems.

From an architectural point of view, industrial networks for production systems are arranged in a hierarchical way as shown in Fig. 3.

Fig. 3 shows three distinct levels: plant, cell and field levels. The upper level or plant level usually implements the overall Supervisory Control And Data Acquisition system (SCADA) [23] for the whole plant. Plants are divided into areas that, in turn, consist of cell groups. Instrumentation and sensor actuation live at the field level. Programmable Logic Controllers (PLC) are organized hierarchically in the network to provide both low level control and data acquisition from the instruments at the field level. The network of master and slave controllers (MC and SC) communicates using industrial Ethernet and serial point-to-point connections with RS232 and RS485 protocols.

An example of a coarse division of such networks is made at the field level [20]. Instrumentation with a 4–20 mA interface uses a point-to-point twisted pair cable. Analog signals are carried over these wires with amplitude modulation. In addition, the widely deployed Highway Addressable Remote Transducer (HART) protocol developed in the 80 s [24] superimposes digital information on the conventional 4–20 mA using a separate digital modulation. DeviceNet, Profibus, AS-I, SDS, Interbus, CANopen or Foundation are examples of the existing range of Fieldbus systems; see [32] for a summary. Fieldbus interface instruments connect to the commercial system with an industrial Ethernet or an RS232/RS485 interface. The standard addresses the problems of monitoring and determining cause and effect of faults in high-speed electro-mechanical, real-time computer controlled processes. Note that Foundation Fieldbus was the only technology prior to wireless technology designed to be control ready and control capable. Foundation Fieldbus is by far the most robust protocol and industrial sensing and control standard built to date. Fieldbus systems offer transmission rates between 125 kbps and 1 Mbps covering distances between 100 and 500 m. Profibus offers higher bandwidth (up to 12 Mbps) and uses messages bigger than 8 bytes as compared with the others [9].