21

NEXUS — Resilient Intelligent Middleware

N Kaveh and R Ghanea Hercock

21.1 Introduction

Service-oriented computing, a composition of distributed-object computing, component-based, and Web-based concepts, is becoming the widespread choice for developing dynamic heterogeneous software assets available as services across a network. One of the major strengths of service-oriented technologies is the high abstraction layer and large granularity level at which software assets are viewed compared to traditional object-oriented technologies. Collaboration through encapsulated and separately defined service interfaces creates a service-oriented environment, whereby multiple services can be linked together through their interfaces to compose a functional system. This approach enables better integration of legacy and non-legacy services, via wrapper interfaces, and allows for service composition at a more abstract level especially in cases such as vertical market stacks. The heterogeneous nature of service-oriented technologies and the granularity of their software components makes them a suitable computing model in the pervasive domain.

Figure 21.1 shows how a service provider advertises its available services (software assets) with a known service broker. This allows entities interested in the services of a service provider, namely service requesters, to query a service broker for the existence and location of such services. Using the obtained broker information, a service requester can make a direct service request to the identified service provider.

![Service-oriented architecture](image_url)
21.2 Motivating Scenario

In this section we present a typical use case scenario in which the NEXUS system is envisaged to be utilised. This scenario highlights the motivation behind this project and serves as a source for the system’s functional and non-functional requirements.

- Situation — a single user needs to perform an information fusion task

An intelligence officer, Jane, is constructing a report on hostile force movements in region X. Due to the textual nature of the task she decides to access NEXUS through a Web browser rather than the NEXUS visual application. This causes the creation of a NEXUS node and agent automatically within the network. Having been authenticated successfully, Jane’s node entry is placed into her assigned group, giving her access to all capabilities and services offered by the other group members.

Once Jane has instructed her agent with the details of the report that she would like compiled, that agent breaks that request into multiple tasks. A task may be carried out locally or it may need to be sent to another agent for processing. For example, most sections of the report require up-to-date field information, which must be requested from their respective remote sources, whereas any textual transformation of the results can be done locally. Five minutes later Jane is presented with a list of the reporting tasks carried out along with their respective results and possible defence data channels in the form of a hyperlinked Web page. She is then in a position to select the most relevant information from the result set for the compilation of the report. One of the defence data channels is a territory image database from which Jane would like to include images. Jane clicks on the link and is provided with the search parameters that can be used for searching and selecting images from the database. Once her search parameters have been provided, her agent takes over the task of connecting, authorising, and fetching relevant images from the database.

Figure 21.2 shows a UML sequence diagram of Jane requesting the retrieval of images based on provided parameters from her agent as discussed in the above scenario. The tasks of locating an appropriate database and retrieving related images is done by the NEXUS agent, transparently from Jane’s viewpoint. Note that the messages shown being passed and the entities are abstract and for modelling the use case. The UML sequence diagram does not contain details of all entities and messages involved; these will be discussed in more detail in the following sections.

The UML sequence diagram in Fig. 21.2 shows Jane’s initial request is automatically broken up by the delegate’s agent and the way in which it interacts with NEXUS to discover services that are required to accomplish Jane’s request. As a new user, Jane would have to register herself by providing general identification details, as well as information to determine her level of authorisation. Jane can then log on to NEXUS via a username and password mechanism (via a network authorisation function) and edit her details at any time. A local repository of user information is then updated to reflect any changes. The main interaction by Jane will be either to initiate a direct search for specific information provided by service(s) on the network (resolved by a discovery function), or to delegate a more general, high-level information retrieval task to her local agent.