

[0016] In the following detailed description, references are made to the accompanying drawings that form a part hereof, and which are shown by way of illustration specific embodiments or examples. Referring now to the drawings, in which like numerals represent like elements throughout the figures, several embodiments and implementations for implementing and utilizing self-describing re-usable software components will be presented.

[0017] While the subject matter described herein is presented in the general context of program modules that execute in conjunction with the execution of an operating system and application programs on a computer system, those skilled in the art will recognize that other implementations may be performed in combination with other types of program modules. Generally, program modules include routines, programs, components, data structures, and other types of structures that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the subject matter described herein may be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, minicomputers, mainframe computers, and the like.

[0018] Turning now to FIG. 1, details will be provided regarding an illustrative operating environment and several software components presented herein. In particular, FIG. 1 shows aspects of a software architecture 100 for providing, discovering, and integrating a re-usable software component 104. As used herein, the term “software component” refers to any type of executable or non-executable software component. By example and not limitation, the re-usable software component 104 may comprise a widget, object, Web part, Web service, data model, workflow, application, plug-in, program fragment, or template. In one particular implementation, the re-usable software component 104 comprises a transformable textually describable data model that is represented using extensible markup language (“XML”) or another markup language.

[0019] The re-usable software component 104 illustrated in FIG. 1 is published in a manner that allows the re-usable software component 104 to be easily discovered and integrated with another software component, such as the application 102. The application 102 is configured for execution on a standard desktop, laptop, or server computer (not illustrated in FIG. 1). In order to permit the application 102 to discover and integrate the re-usable software component 104, knowledge elements 106 are published for use with the re-usable software component 104.

[0020] The knowledge elements 106 comprise data that identifies the capabilities and affinities of the re-usable software component 104 along with one or more mechanisms for integrating the re-usable software component 104 with other software components. The knowledge elements 106 may be specified at any time prior to publication or at the time of publication of the re-usable software component 104. As used herein, the term “publication” refers to the process of making the re-usable software component 104 available to re-users. The re-usable software component 104 is “self-describing” in that the data set forth by the knowledge elements 106 specifies the capabilities, affinities, and integration mechanisms for the re-usable software component 104.

[0021] According to various embodiments, the knowledge elements 106 comprise data that is formatted using XML, another markup language, or another data format altogether,

that identifies the capabilities of the re-usable software component 104. According to embodiments, the knowledge elements 106 also identify one or more other software components with which the re-usable software component 104 may be integrated. In this way, the knowledge elements 106 indicate the “affinities” of the re-usable software component 104. The capabilities identified by the knowledge elements 106 may be scoped to one or more of the other software components that the re-usable software component 104 may be integrated with.

[0022] According to other embodiments, the knowledge elements 106 also contain data that identifies one or more mechanisms for integrating the re-usable software component 104 with other software components. The mechanisms for integrating the re-usable software component 104 may be scoped for use with a particular other software component or set of other software components. The mechanisms for integrating the re-usable software component 104 may also be scoped to one or more of the capabilities of the re-usable software component 104 identified by the knowledge elements 106.

[0023] In one embodiment where the re-usable software component 104 comprises a transformable textually describable model, the mechanisms identified by the knowledge elements 106 for integrating the re-usable software component 104 with other software components may include one or more data transformations to be performed in order to integrate the re-usable software component 104. In this way, the knowledge elements 106 can specify how the re-usable software component 104 is to be transformed to allow it to integrate with other software components, like the application 102. For instance, the knowledge elements 106 may specify a merge operation or a union operation that is to be performed between the re-usable software component 104 and another software component to integrate the two components.

[0024] As will be discussed in greater detail below, the knowledge elements 106 may be searched to discover an appropriate re-usable software component 104 for use in a particular application. In this regard, it should be appreciated that although only a single re-usable software component 104 has been illustrated in FIG. 1, many re-usable software components 104 and their corresponding knowledge elements 106 may be published in a library, database, or in another manner, and made available for consumption by re-users. Once an appropriate re-usable software component 104 has been discovered, the knowledge elements 106 are also utilized to integrate the re-usable software component 104 in a fully- or semi-automated fashion. Additional details regarding publication of the knowledge elements 106, and the discovery and integration of a re-usable software component 104 are provided below.

[0025] Although FIG. 1 illustrates the integration of an application 102 with the re-usable software component 104, it should be appreciated that the embodiments presented herein permit the integration of any type of software component with a re-usable software component 104. It should also be appreciated that although the knowledge elements 106 are illustrated in FIG. 1 as being located external to the re-usable software component 104, the knowledge elements 106 may also be stored within the re-usable software component 104 itself. The knowledge elements 106 may also be stored separately as shown in FIG. 1 and made available through a database or other mechanism. When stored separately, the