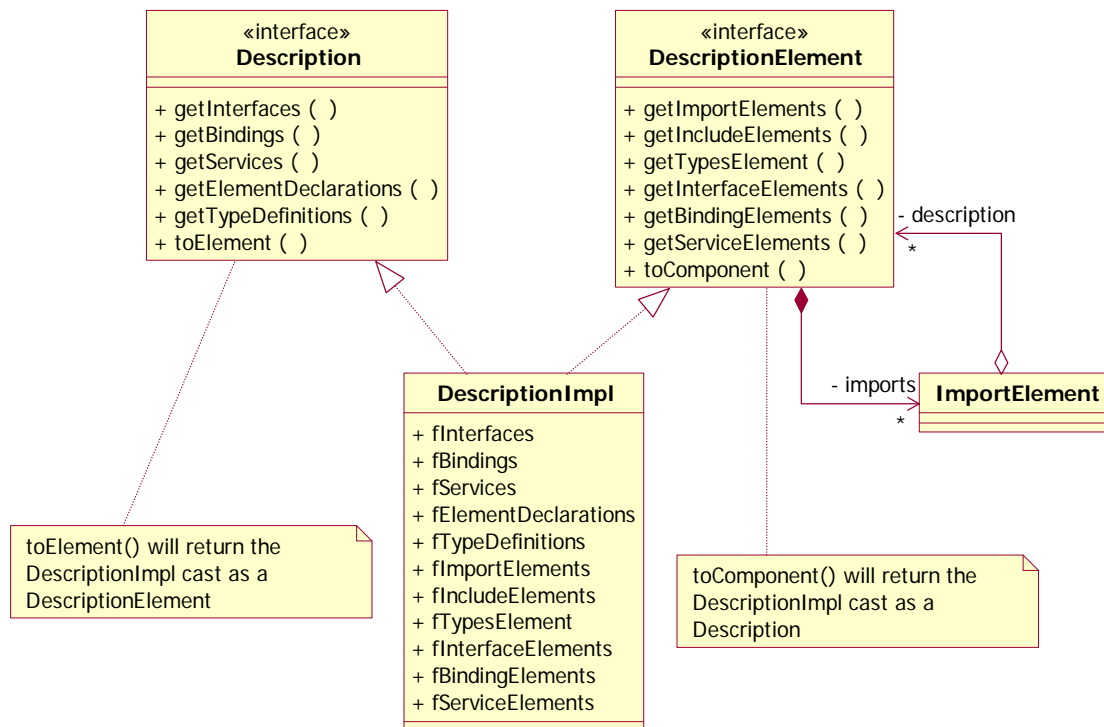


Design for WSDL Description

The **Description** interface represents the Description component from the abstract component model. The **DescriptionElement** interface represents the <wsdl:description> element.

These were initially implemented by separate classes, **DescriptionImpl** and **DescriptionElementImpl**, on the basis that Description was a 'fatted view' container for components potentially from a tree of WSDL documents while DescriptionElement represented just one level of WSDL. In hindsight this seems an unnecessary distinction. A DescriptionElement is also a container for imported or included WSDL, it just exposes the structure and makes this concrete rather than abstract.

The design has been changed so that now both interfaces are implemented by a single class, DescriptionImpl, following the same approach used for the other WSDL component and element interfaces. The DescriptionElementImpl class will be deleted, pending confirmation of this design change.



Note that for a <description> element describing just a fragment of a web service, let's say the <types> and <interface> only, the toComponent() method will return a Description component that has no Binding or Service components (i.e. it does not fully describe a web service). This doesn't seem particularly useful given that the abstract model has no concept of combining Description components. However, it does conform with the WSDL 2.0 Part 2 spec which states that all of the top-level components contained in the Description component (e.g. Interface, Binding, Service) are optional.

Note also that the toElement() and toComponent() methods could be moved to a separate class (e.g. Reader or some converter/builder class) to eliminate any cross-references between the Component and Element interfaces. Consideration of this has been marked as a TODO.