

8. Possible use of IDL

[From Wikipedia]

An interface description language (or alternately, interface definition language), or *IDL* for short, is a computer language used to describe a software component's interface. IDLs describe an interface in a language-neutral way, enabling communication between software components that do not share a language – for example, between components written in C++ and components written in Java.

IDLs are commonly used in remote procedure call software. In these cases the machines at either end of the "link" may be using different operating systems and computer languages. IDLs offer a bridge between the two different systems.

For OpenMI I see two possible uses:

1. Use IDL to define the OpenMI Standard (interfaces) in a language-neutral format, and provide mappings for languages supported by OATC (e.g. CLS and Java). This would make OpenMI more generic and "open the door" for future implementations in other programming languages, at the cost of added complexity.
2. Use IDL and defined mappings for CLS and Java for interoperability between models (using IIOP.Net and Java RMI/IIOP).