NEBULON FDD INTERCHANGE (FDDI) SPECIFICATION

October 27 2005

Patent

The attention of adopters is directed to the possibility that compliance with or adoption of Nebulon FDDI specifications may require use of an invention covered by patent rights. Nebulon shall not be responsible for identifying patents for which a license may be required by any Nebulon specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. Nebulon FDDI specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

Notice

The information contained in this document is subject to change without notice. The material in this document details a Nebulon specification in accordance with the license and notices set forth on this page. This document does not represent a commitment to implement any portion of this specification in any company's products.

Issue Reporting

All Nebulon FDDI specifications are subject to continuous review and improvement. As part of this process we encourage readers to report any ambiguities, inconsistencies, or inaccuracies they may find.

TABLE OF CONTENTS

PREFACE

About the FDDI Specification

The main purpose of FDDI is to enable the exchange of FDD project related information between diverse software systems and components.

FDDI (**f ĭ d ˈ ē**) is pronounced fiddy (as for the word giddy but replacing the g with an f).

Intended Audience and Use

This first public draft is for FDD tool authors and other interested parties to analyse, discuss and provide feedback at http://www.featuredrivendevelopment.com

The scope in this draft is to allow you to communicate enough information to produce the "standard" reports that are mentioned in the FDD documentation. It doesn't cater for the trend report, as the whole notion of reporting over time is a complex one. The issues are understood, but it was felt that it added too much complexity for this first draft.

STRUCTURE

Core

The “core” revolves around a project with the following structure.

Project

    Subject*

        Activity*

            Feature*

                Milestone*

This might look familiar: it’s basically a hierarchical list of features, with 2 levels of breakdown, along with milestones. This is the structure described in the original JMCU book, and in some detail at the website www.featuredrivendevelopment.com as well.

In the past this structure was known as a project, but that’s not what it’s really about. On a typical FDD project we would have one of these structures for the PD layer, one for the UI layer, and possibly one for the SI layer. All of these together are “the project”.

However, this decomposition of the project is not always along architectural boundaries. We might expose other parts of the project, such as documentation (e.g. writing the online help), testing, or externally subcontracted work, through one of these structures.

We now call one of these structures a project aspect.

So now a project has some number of aspects, each having its own breakdown of structures. Note that each aspect may have different milestones, for example it’s typical for PD and UI to use a different set of milestones.

We also want to allow the aggregation of projects in some way. Program management is typically how this is done, so we provide a simple hierarchical organisation of project as well.

So now the structure becomes:

Program

Project*

Aspect*

AspectDesc?

MilestoneDesc+

Subject*

Activity*

Feature*

Milestone*

Progress

We also want to allow the exchange of progress information, not just the project structure. Fortunately the information we want to report is pretty much the same at all levels, whether it's a business activity, subject area, of the project as a whole.

An individual feature has a percentage completion figure, derived from its milestones, and an overall status. For each level of aggregation above this we record the rolled up percentage completion and the total number of features. We also breakdown the feature counts by status, for example "completed" vs. "not started". Optionally we can also derive an overall status at the rolled up level.

This gives a progress element that looks something like this:

```
<progress count="43" completion="57" status="underway">
    <kpi status="notstarted" count="0"/>
    <kpi status="underway" count="3"/>
    <kpi status="attention" count="0"/>
    <kpi status="complete" count="40"/>
    <kpi status="inactive" count="0"/>
</progress>
```

This element can be attached to a program, project, subject or activity to provide rolled-up summary data. Since the count attribute defaults to 1, and the kpi's are optional, we can reuse the same element to describe the progress of a feature as well. For milestones we can optionally attach a status attribute.

Extension

Two facilities have been added to allow for schema extensions. The first is that any element in the fddi schema can have additional attributes, as long as they are from a different namespace. The second is the use of extension elements, in a similar fashion to XMI. For most elements we can specify a number of extensions, with arbitrary content, that are intended for use by third party tools.

SCHEMA

```
<?xml version="1.0" encoding="UTF-8" ?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                    targetNamespace="http://www.nebulon.com/xml/2004/fddi"
                    xmlns:fddi="http://www.nebulon.com/xml/2004/fddi">

 <xsd:annotation>
  <xsd:documentation>
   The FDD Interchange (FDDI) Schema.
  </xsd:documentation>
 </xsd:annotation>

 <xsd:attributeGroup name="baseAttrs">
  <xsd:annotation>
   <xsd:documentation>
    Attributes common to all elements. Any element can have an optional
    "id" attribute.

      The extension mechanism allows us to place arbitrary attributes on any
    element, but they must come from a different namespace.
   </xsd:documentation>
  </xsd:annotation>

  <xsd:attribute name="id" type="xsd:ID" use="optional"/>
  <xsd:anyAttribute namespace="##other" processContents="lax"/>
 </xsd:attributeGroup>

 <xsd:element name="fddi">
  <xsd:annotation>
   <xsd:documentation>
    Top level container for the XML fragment.

    This can contain some number of either programs, projects or aspects
    (but cannot be a mixture of these) and any number of extension
    elements.
   </xsd:documentation>
  </xsd:annotation>

  <xsd:complexType>
   <xsd:sequence>
    <xsd:choice>
     <xsd:element ref="fddi:program" minOccurs="0" maxOccurs="unbounded" />
     <xsd:element ref="fddi:project" minOccurs="0" maxOccurs="unbounded" />
     <xsd:element ref="fddi:aspect" minOccurs="0" maxOccurs="unbounded" />
```

```
    </xsd:choice>
    <xsd:element name="extension" type="fddi:extension" minOccurs="0" maxOccurs="unbounded" />
   </xsd:sequence>

   <xsd:attributeGroup ref="fddi:baseAttrs"/>
  </xsd:complexType>
</xsd:element>


<xsd:element name="program">
  <xsd:annotation>
   <xsd:documentation>
    Programs are a way to aggregate Projects into a hierarchy. A program may
    contain either some number of other programs or some number of projects
    but not both.

    In additional it may contain an optional progress element and any
    number of extension elements.
   </xsd:documentation>
  </xsd:annotation>

  <xsd:complexType>
   <xsd:sequence>
    <xsd:choice>
     <xsd:element ref="fddi:program" minOccurs="0" maxOccurs="unbounded" />
     <xsd:element ref="fddi:project" minOccurs="0" maxOccurs="unbounded" />
    </xsd:choice>
    <xsd:element name="progress" type="fddi:progress" minOccurs="0" maxOccurs="1" />
    <xsd:element name="extension" type="fddi:extension" minOccurs="0" maxOccurs="unbounded" />
   </xsd:sequence>

   <xsd:attributeGroup ref="fddi:baseAttrs"/>
   <xsd:attribute name="name" type="xsd:token" use="required" />
  </xsd:complexType>
</xsd:element>

<xsd:element name="project">
  <xsd:annotation>
   <xsd:documentation>
    A project is some number of aspects, where each aspect may be of a
    different type. For example: UI, PD and SI aspects. It may also contain
    an optional progress element and any number of extension elements.
   </xsd:documentation>
  </xsd:annotation>

  <xsd:complexType>
```

```
    <xsd:sequence>
      <xsd:element ref="fddi:aspect" minOccurs="0" maxOccurs="unbounded" />
      <xsd:element name="progress" type="fddi:progress" minOccurs="0" maxOccurs="1" />
      <xsd:element name="extension" type="fddi:extension" minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>

    <xsd:attributeGroup ref="fddi:baseAttrs"/>
    <xsd:attribute name="name" type="xsd:token" use="required" />
  </xsd:complexType>
</xsd:element>

<xsd:element name="aspect">
  <xsd:annotation>
    <xsd:documentation>
      Aspects are a "Feature Breakdown Structure" (FBS), which are a two-level
      decomposed list of features and optionally the non-standard milestone
      definitions to be used.

      It may also contain an optional progress element and any number of
      extension elements.
    </xsd:documentation>
  </xsd:annotation>

  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="description" type="fddi:aspectDesc" minOccurs="0" maxOccurs="1" />
      <xsd:element name="subject" type="fddi:subject" minOccurs="0" maxOccurs="unbounded" />
      <xsd:element name="progress" type="fddi:progress" minOccurs="0" maxOccurs="1" />
      <xsd:element name="extension" type="fddi:extension" minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>

    <xsd:attributeGroup ref="fddi:baseAttrs"/>
    <xsd:attribute name="name" type="xsd:token" use="required" />
  </xsd:complexType>
</xsd:element>

  <xsd:complexType name="aspectDesc">
    <xsd:annotation>
      <xsd:documentation>
        An aspect description is the set of milestone definitions and the
        names of key concepts: subject area, business activity, feature,
        milestone.
      </xsd:documentation>
    </xsd:annotation>

    <xsd:sequence>
```

```
      <xsd:element name="milestoneDesc" type="fddi:milestoneDesc" minOccurs="1" maxOccurs="unbounded" />
      <xsd:element name="extension" type="fddi:extension" minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>

    <xsd:attributeGroup ref="fddi:baseAttrs"/>
    <xsd:attribute name="subjectName" type="xsd:token" />
    <xsd:attribute name="activityName" type="xsd:token" />
    <xsd:attribute name="featureName" type="xsd:token" />
    <xsd:attribute name="milestoneName" type="xsd:token" />
  </xsd:complexType>


  <xsd:complexType name="milestoneDesc">
    <xsd:annotation>
      <xsd:documentation>
        A milestone is defined by a name and effort, and may optionally
        contain any number of extension elements.
      </xsd:documentation>
    </xsd:annotation>

    <xsd:sequence>
      <xsd:element name="extension" type="fddi:extension" minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>

    <xsd:attributeGroup ref="fddi:baseAttrs"/>
    <xsd:attribute name="name" type="xsd:string" use="required" />
    <xsd:attribute name="effort" type="xsd:nonNegativeInteger" use="required" />
  </xsd:complexType>

  <xsd:complexType name="subject">
    <xsd:annotation>
      <xsd:documentation>
        This is the top level of the FBS, named "Subject Area" in a
        standard project.
      </xsd:documentation>
    </xsd:annotation>

    <xsd:sequence>
      <xsd:element name="activity" type="fddi:activity" minOccurs="0" maxOccurs="unbounded" />
      <xsd:element name="progress" type="fddi:progress" minOccurs="0" maxOccurs="1" />
      <xsd:element name="extension" type="fddi:extension" minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>

    <xsd:attributeGroup ref="fddi:baseAttrs"/>
    <xsd:attribute name="prefix" type="xsd:token" use="required" />
    <xsd:attribute name="name" type="xsd:token" use="required" />
```

```
</xsd:complexType>

<xsd:complexType name="activity">
  <xsd:annotation>
   <xsd:documentation>
    This is the second level in the FBS hierarchy, named "Business
    Activity" in a standard project.
   </xsd:documentation>
  </xsd:annotation>

  <xsd:sequence>
   <xsd:element name="feature" type="fddi:feature" minOccurs="0" maxOccurs="unbounded" />
   <xsd:element name="progress" type="fddi:progress" minOccurs="0" maxOccurs="1" />
   <xsd:element name="extension" type="fddi:extension" minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>

  <xsd:attributeGroup ref="fddi:baseAttrs"/>
  <xsd:attribute name="name" type="xsd:token" use="required" />
  <xsd:attribute name="target" type="xsd:gYearMonth" use="optional" />
  <xsd:attribute name="initials" type="xsd:NCName" use="optional" />
</xsd:complexType>

<xsd:complexType name="feature">
  <xsd:annotation>
   <xsd:documentation>
    A feature is some small piece of client valued function.
   </xsd:documentation>
  </xsd:annotation>

  <xsd:sequence>
   <xsd:element name="name" type="xsd:string" />
   <xsd:element name="milestone" type="fddi:milestone" minOccurs="0" maxOccurs="unbounded" />
   <xsd:element name="remarks" type="fddi:note" minOccurs="0" maxOccurs="unbounded" />
   <xsd:element name="progress" type="fddi:progress" minOccurs="0" maxOccurs="1" />
   <xsd:element name="extension" type="fddi:extension" minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>

  <xsd:attributeGroup ref="fddi:baseAttrs"/>
  <xsd:attribute name="seq" type="xsd:positiveInteger" use="required" />
  <xsd:attribute name="initials" type="xsd:string" />
</xsd:complexType>

<xsd:complexType name="note" mixed="true">
  <xsd:annotation>
   <xsd:documentation>
    Notes may contain the initials of who wrote them, and the date
```

```
        entered.

        In addition they may contain extension elements in a mixed content
        model and the usual base attributes.
      </xsd:documentation>
    </xsd:annotation>

    <xsd:sequence>
      <xsd:element name="extension" type="fddi:extension" minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>

    <xsd:attributeGroup ref="fddi:baseAttrs"/>
    <xsd:attribute name="entered" type="xsd:date" use="optional" />
    <xsd:attribute name="initials" type="xsd:string" use="optional" />
  </xsd:complexType>

  <xsd:complexType name="milestone">
    <xsd:annotation>
      <xsd:documentation>
        A milestone has a planned date, an optional actual date, and maybe
        a derived status as well. It may optionally contain any number of
        extension elements as well.
      </xsd:documentation>
    </xsd:annotation>

    <xsd:sequence>
      <xsd:element name="extension" type="fddi:extension" minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>

    <xsd:attributeGroup ref="fddi:baseAttrs"/>
    <xsd:attribute name="planned" type="xsd:date" use="required" />
    <xsd:attribute name="actual" type="xsd:date" use="optional" />
    <xsd:attribute name="status" type="fddi:statusEnum" use="optional" />
  </xsd:complexType>

 <xsd:simpleType name="statusEnum">
  <xsd:annotation>
    <xsd:documentation>
      An enumerated type for "status" attributes, used on milestones and
      progress elements.
    </xsd:documentation>
  </xsd:annotation>

  <xsd:restriction base="xsd:NMTOKEN">
    <xsd:enumeration value="notstarted"/>
    <xsd:enumeration value="underway"/>
```

```
    <xsd:enumeration value="attention"/>
    <xsd:enumeration value="complete"/>
    <xsd:enumeration value="inactive"/>
  </xsd:restriction>
</xsd:simpleType>

  <xsd:complexType name="progress">
    <xsd:annotation>
      <xsd:documentation>
        A snapshot of the progress of "something": program, project, aspect,
        subject, activity, or feature.
      </xsd:documentation>
    </xsd:annotation>

    <xsd:sequence>
      <xsd:element name="kpi" minOccurs="0" maxOccurs="unbounded" >
        <xsd:complexType>
          <xsd:attribute name="status" type="fddi:statusEnum" use="required" />
          <xsd:attribute name="count" type="xsd:nonNegativeInteger" use="required" />
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="extension" type="fddi:extension" minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>

    <xsd:attributeGroup ref="fddi:baseAttrs"/>
    <xsd:attribute name="count" type="xsd:nonNegativeInteger" default="1" />
    <xsd:attribute name="completion" type="xsd:nonNegativeInteger" use="required" />
    <xsd:attribute name="status" type="fddi:statusEnum" />
  </xsd:complexType>

  <xsd:complexType name="extension">
    <xsd:annotation>
      <xsd:documentation>
        Most elements in the FDDI schema may contain any number of "extension"
        elements. This is to allow additional, arbitrary data to be attached
        to elements for purposes not specified by the interchange standard.

        The contents of the extension elements is undefined. An "extender"
        attribute must be present to hopefully ensure different extension
        mechanisms do not clash with each other. What values should be used
        to achieve this is not specified in this schema definition.
      </xsd:documentation>
    </xsd:annotation>

    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:any processContents="lax"/>
```

```
    </xsd:choice>

    <xsd:attribute name="extender" type="xsd:string" use="required"/>
    <xsd:attribute name="extenderID" type="xsd:string" use="optional"/>
    <xsd:anyAttribute namespace="##other" processContents="lax"/>
  </xsd:complexType>

</xsd:schema>
```