



Enabling Grids for E-scienceE

NPM NM-WG Object Model

JRA4 F2F, Edinburgh, 12-13 July 2005

Alistair K Phipps (A.Phipps@ed.ac.uk)

University of Edinburgh

www.eu-egee.org



Information Society



INFSO-RI-508833

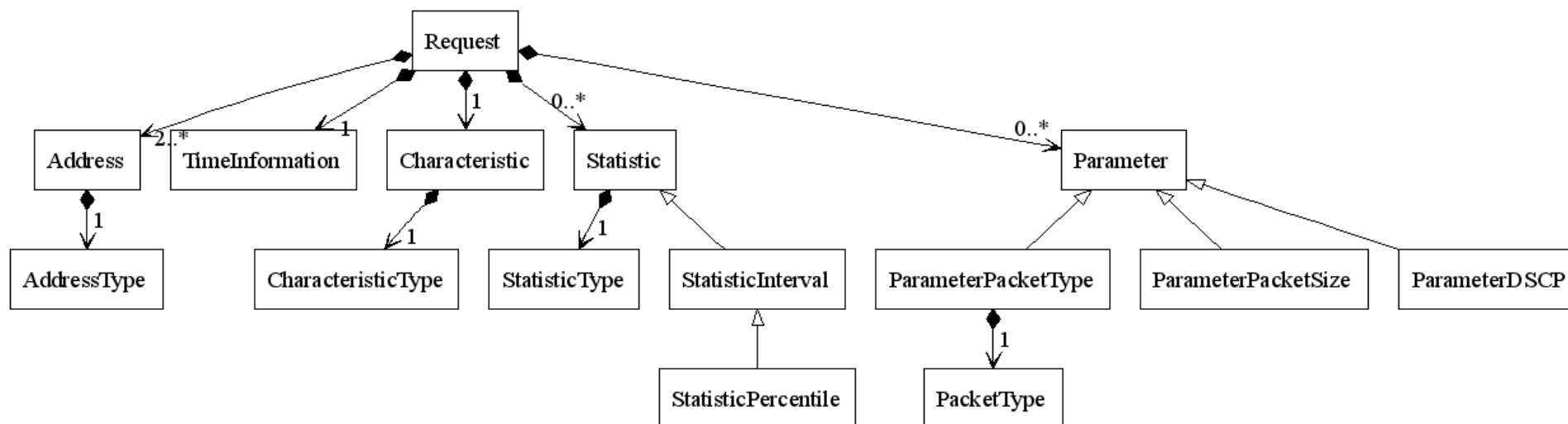
- **NM-WG interface is defined as an XML schema**
- **Two alternatives for writing software to adhere to this interface:**
 - Directly parse XML using an XML parser library
 - Writing XML parsing code tricky and prone to errors
 - Use software such as Apache Axis to generate "stub" classes
 - Stub classes mirror the schema elements and carry out the XML parsing internally (and automatically)
 - However, they are automatically generated and are often quirky and awkward to use

- **What is it?**
 - A set of objects that encapsulate part of the NM-WG schema (just what we use) in a sensible way
 - A 'mapper' to translate these objects into the form of Axis stub objects (though the mapper could also do direct XML parsing)
 - Designed with reference to the draft schema
- **Who is it useful for?**
 - Implementers of services and clients that have to 'talk' the schema
- **Why not just use Axis stubs directly?**
 - Would tightly bind the 'business logic' to the interface; instead, the business logic uses the object model objects which do not vary with (minor, at least) interface changes - only the mapper changes
 - As noted previously, the Axis stubs are generated automatically and are awkward to use

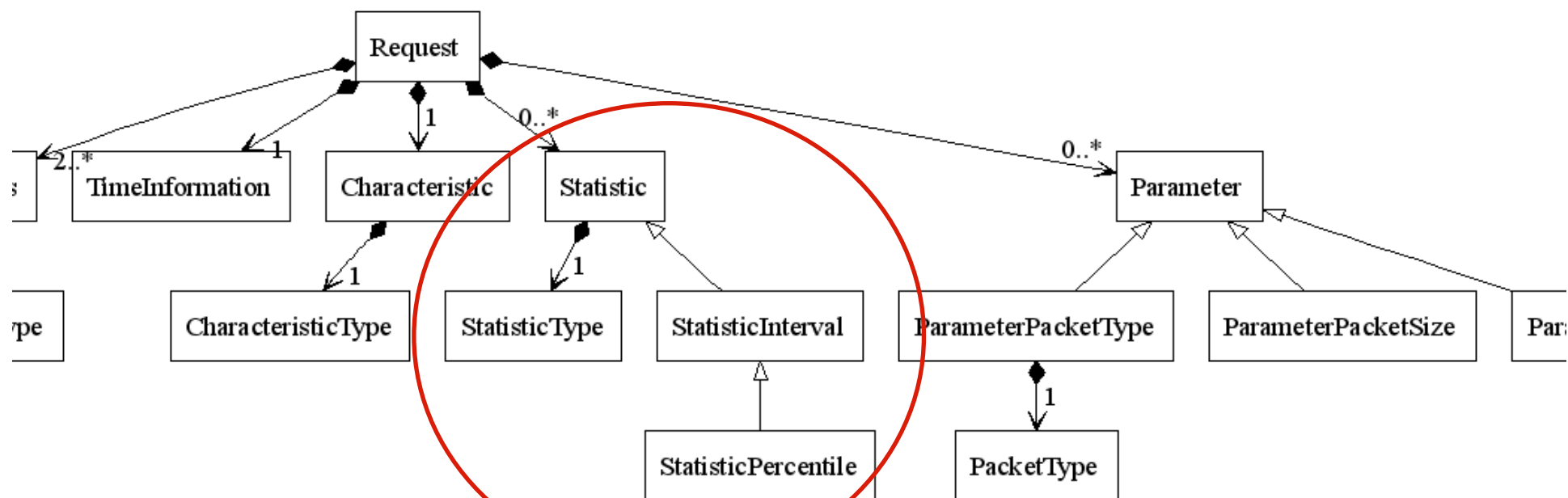
- **Covers:**
 - NM-WG features covered by the object model in the draft design
 - Design overview (for those implementing components that will be making use of the object model)
- **Objectives:**
 - Reach agreement on the NM-WG (v1 and v2) features covered by the object model, e.g.:
 - Do we need to request or report packet gap information?
 - Do we need separate classes for the different types of result (e.g. AvailableBandwidth including bottleneck information)?
 - Gather design improvements / corrections

- **Current status:**
 - Draft design document written:
<http://edms.cern.ch/document/606702/>
 - Implementation of this to happen soon after F2F
- **Will not go through entire design in detail - instead give overview, then draw attention to specific areas which merit discussion**

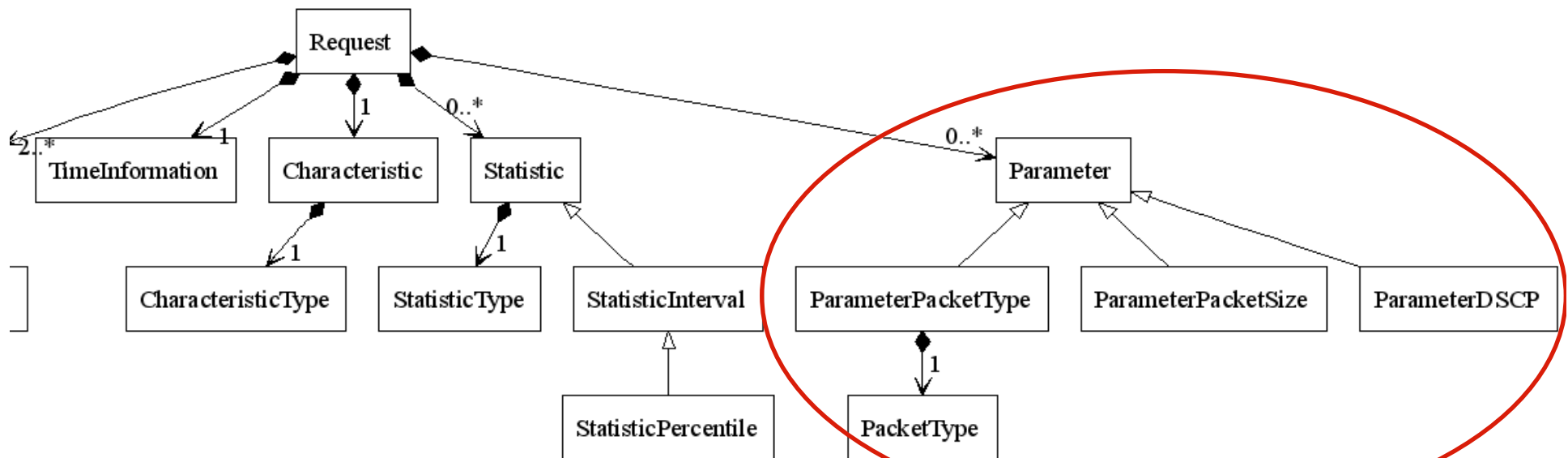
- Request has:
 - List of hop addresses. Addresses include IP/hostname, type (IPv4, IPv6), optional 'friendly' name (string)
 - TimeInformation. Time focus, optional lower and upper bounds.
 - Characteristic. Includes enumerated characteristic type. (e.g. ONE_WAY_DELAY)
 - List of Statistics
 - Set of Parameters



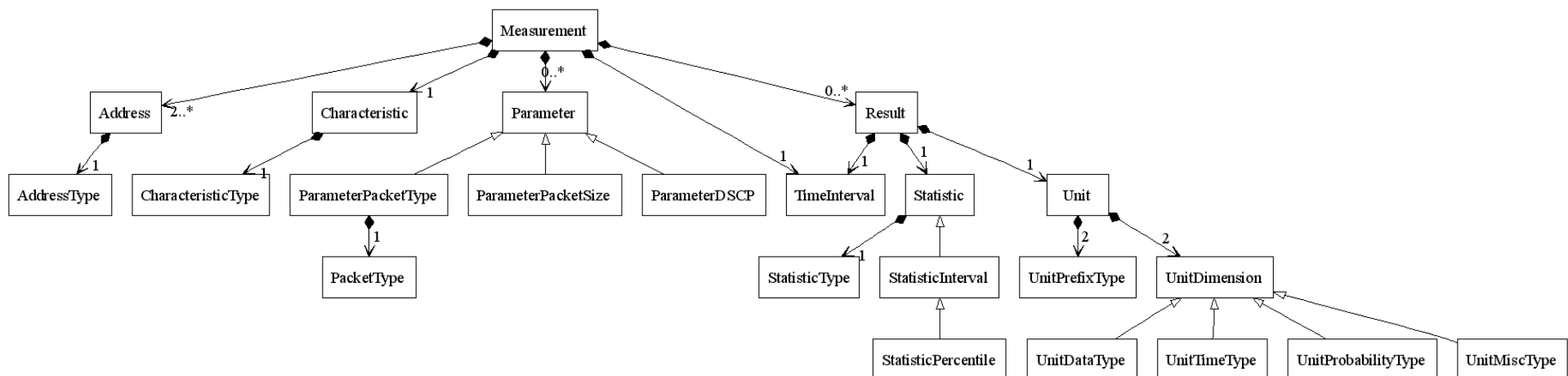
- **Statistic** includes an enumerated **StatisticType** (e.g. MAX)
- **StatisticInterval** specialisation also includes a time interval over which the statistic should be applied, in milliseconds
- **StatisticPercentile** includes a time interval and also a percentile value, in percent



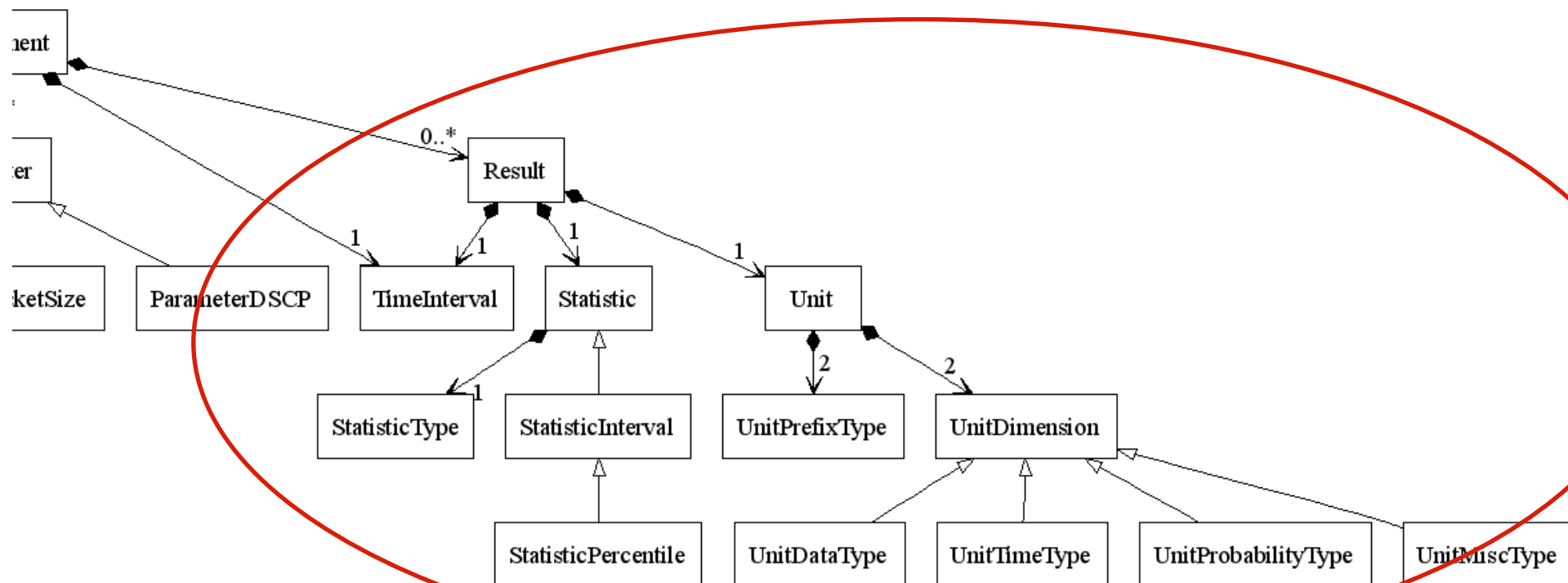
- Parameter includes whether the parameter is required
- ParameterPacketType specialisation also includes a packet type (TCP, UDP, ICMP)
- ParameterPacketSize specialisation also includes a packet size (in bytes)
- ParameterDSCP specialisation also includes a DSCP value byte



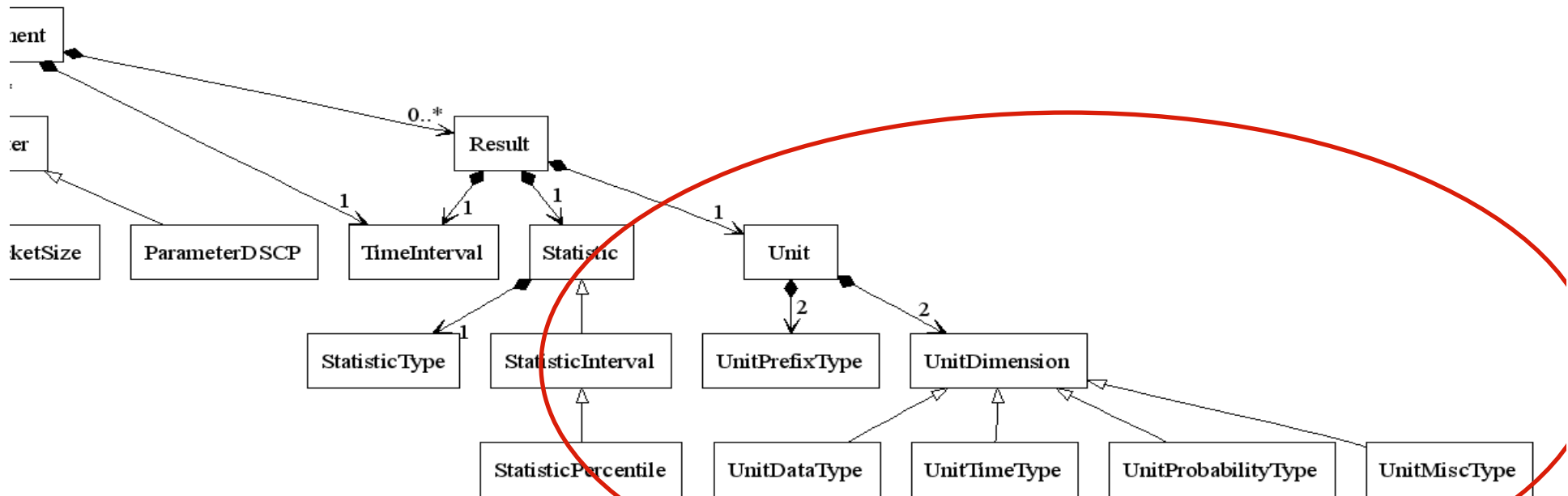
- A Report consists of an array of Measurements; NM-WG object model does not encapsulate the Report explicitly
- Measurement has:
 - List of hop addresses, Characteristic, Set of Parameters
 - TimeInterval - lower and upper bounds on time for the whole measurement
 - List of Results
 - List of tools (strings)



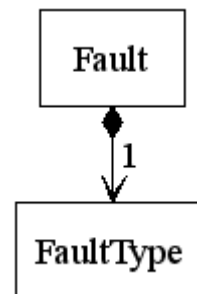
- **Result has:**
 - Value (double)
 - Unit
 - Statistic (as for Request)
 - TimeInterval - upper and lower bound on time for this specific Result



- **Unit has:**
 - Numerator and denominator unit (e.g. byte, bit, second)
 - Numerator and denominator prefix (e.g. kilo, micro, kebi)
- **Supports conversions between Units and to/from strings**



- **Fault has:**
 - Enumerated fault type
 - Fault message
- **Extends (Java) Exception - can be thrown**



- A mapper class translates between NM-WG object model entities and NM-WGv1 stubs generated by Axis
- It should be possible to write similar mappers to translate to NM-WGv2 stubs (GN2:JRA1 TL?)
- AxisMapper is a singleton (only one instance created) that provides methods to convert back and forth, e.g.:
 - `axis.report.NetworkMeasurementReport measurementArrayToAxisNetworkMeasurementReport(Measurement[] measurementArray)`
 - `Measurement[] axisNetworkMeasurementReportToMeasurementArray(axis.report.NetworkMeasurementReport report)`

- **Faults: any more?**
 - System
 - No Data
 - Communication
 - Authorisation
 - Request
- **Units: any more?**
 - Only "unit" or "unit1/unit2" permitted
 - Data: bit, byte
 - Time: second, minute, hour
 - Probability: percent, probability
 - Misc: none
 - Prefixes: micro to tera (steps of 1000), kibi to tebi (steps of 1024)

- **Parameters - any more?**
 - Packet Type
 - Packet Size
 - DSCP
- **Packet types - any more?**
 - TCP
 - UDP
 - ICMP
- **Address types - any more?**
 - IPv4
 - IPv6

- **Statistics - any more?**
 - Raw
 - Count
 - Min
 - Max
 - Mean
 - Median
 - Standard deviation
 - Percentile

- **Characteristics - any more?**
 - Path delay one-way
 - Path delay round-trip
 - Path bandwidth achievable
 - Path loss one-way
 - Path loss round-trip
 - Path bandwidth capacity
 - Hop bandwidth utilised
 - Path delay one-way jitter
 - Path packet reordering

- Any other issues? Questions? Comments?