

Minutes of the Analysis & Design Meeting

Date: 11/01/2000
Present: Nikolai, Frank, Kris, Vito, Alessandro
Agenda: The definition of the data pushed by the RT to the MT for a property subscription

Abbreviations: GT = GUI Tier/ MT = Middle Tier/ RT = Resource Tier

Type of Data Provided by the Device Server in the RT

We first recall that any data acquired by a device server in the RT can be tagged either as **new data** or as **data changed**.

Data tagged with “data changed” flag is “new data” where the value, compared with the previous acquired one, has changed according to any rule implemented by the device server.
Data tagged with the “data changed” flag can be used to simplify the subscription on change in the MT.

A subscription to “new data” can be useful for someone who wants to implement a polling system by reusing the subscription mechanism.

The Structure of the Data Packet

We have identified the list of information that has to be sent by the RT to the MT.

The packet will contain at least the following information:

Data Value	Date	Cycle Stamp	Data Changed Flag	Data Quality	
------------	------	-------------	-------------------	--------------	------

Example of field values.

- **Data Value:** any value
- **Date:** the (POSIX ?) date registered when the data value has been acquired by the device server
- **Cycle Stamp:** a value representing the “SuperCycle+Cycle+CycleInstance” registered when data has been acquired (to be discussed with timing people)
- **Data Changed Flag:** 0=new data/ 1=data changed
- **Data Quality:** 0=OK/ 1=ERROR/ 2=....

Errors

Valid data and errors are not exclusive and they may coexist. For example, if we send a complex data with information that are partially correct, an error can be there and specify where is the problem or which data are valid.

The error flag in the Data Quality field is not related to the status of the device, which is accessed with dedicated properties, but only to the middleware environment.

An error can be fully qualified with an extra field: for example, Data Quality= 1 (error) and Error Type= 22, where 22=“Device Server Busy”. This is the present PS general mechanism for transporting errors from RT to GT.

Mandatory Data

Except for the Cycle Stamp, which must be defined only for the cycle dependent devices, all fields in the data packet structure are mandatory.

Implementation of the Data Packet

We have identified two ways to implement such a structure:

- The first one consists in creating a base class with all mandatory fields. Any extension will be done by sub classing the base class. Pros: flexible solution. Contras: we have to deal with many user-defined subclasses and can be difficult to see if information is present or not.
- The second way for implementing the structure is to define a limited number of standard tagged messages. For example, one for the normal message, another one for the error message and so on. Pros & Contras are the negation of the previous ones.

The two implementation are not mutually exclusive.

Nikolai and Kris prefer the second implementation. Vito the first one.

The decision on how to implement the message structure has been postponed after the selection of the MOM. This is important for throughput optimisation.

Filtering

The filtering at the RT level should be minimal (Vito prefers no filtering at all).

Every property subscribed by a GT client should be identified with **Device+Property+Selector**.

The selector has to be specified, but should be something like

“SuperCycle+Cycle+CycleInstance” (to be discussed with timing people). The subscription information will be propagated at every level (MT+RT).

The selector will allow, but not impose, to reduce the network traffic from the RT to the MT.

We all agreed that, in the first prototype, we will not optimise the traffic in the RT but only in the MT.

In the second version the RT will eventually optimise the traffic over the network.

Actions

Meet again next Tuesday (18/01/00) and finalize the structure of the packet (IDL def.) and definition of the interface.

Alessandro