Protocol PR2014-12-17
Software Design of "Interface"

Author: M. Siegl, G. Damianitsch
File: PR2014-12-17
Modified: 2014-12-27
Version: 1.1

1 Background

In PR2014-12-02 chapter 2.3 the communication between DataPoints and Buffers defined.

1. *The transfer of information between Buffer and DataPoint goes over a socket. The socket is defined for communication via the IP-Network. Therefore a distributed "Interface" can be created later.*
2. *The communication is coded in XML. The associated definition (XSD) must be defined.*

This document describes details of the communication design.

2 Results of Discussion

2.1 Java Virtual Maschines

1. The focus lays on the Java Virtual Maschines, which hold parts of the application on the G2021. It is not of relevance, who many JVMs are running on one G2021 and it is not of relevance, who many G2021 are in action to run the JVMs. It is possible, all JVMs running on one single G2021. It is possible, each G2021 holds only one JVM.
2. All Buffers, available on one G2021, are running inside one single JVM on this G2021. A G2021 can hold zero or one JVM with Buffers.
3. One GBot has one DataPoint. One or more GBot/DataPoint are running on a JVM. A G2021 can hold zero or one or more then one JVM with Gbot/DataPoint.
4. Buffers and GBot/DataPoints are not running on the same JVM.

2.2 Low Level Communication

1. The transport runs via a socket and TCP/IP. There are Comm-Servers and Comm-Clients. A JVM running the Buffers, holds a Comm-Server. A JVM running GBots/DataPoints holds a Comm-Client. Comm-Server and Comm-Client are not running on the same JVM.
2. Communication is based in Server-Client mechanism.
3. Communication is based on telegrams. The telegrams are formated as XML-container. The XML-container are separated by the character-sequence `\r\n\r\n`.
4. The telegrams are grouped in request-telegrams and response-telegrams. A request-telegram can be asynchronous sent by the Comm-Server or the CommClient. Each request-telegram is followed by a response-telegram in the opposite direction.
5. An exception exists: when the Comm-server or the Comm-client wants to disconnect the

connection, it sends a Disconnect-telegram. A Disconnect-telegram does not have a response-telegram. The TCP/IP-connection will be terminated immediatelly.

6. Because of the usage of TCP/IP, in the higher level protocols there is no need for error handling or recovering of telegrams.

2.3 XML Communication

This section describes the functionality of telegrams. The used names are not the names of the according classes (which handles the telegrams).

2.3.1 Connection Management

Some telegrams are required to manage a connection. Each implementation of the communication is signed by a individual version number. The Comm-client requests communication based on a specific version number. The Comm-server must accept the requestes version or it has to reject the connect-request. The disconnect can be requested by the Comm-server as well as the Comm-client at any time.

1. Basic-setup-connect (comm-protocol version)
2. Basic-setup-accepted ()
3. Basic-setup-reject ()
4. Basic-setup-disconnect ()

2.3.2 Process Communication Query

After a connection is setup, the Comm-client can request some information about available buffers. Each buffer has assigned an individual name. The name must be unique. Details about naming has to be defined. Each buffer has assigned some meta-data. The meta-data are structured similar to properties in java – paires of *name=value*.

If a GBot need values from a buffer, it searches using regular-expressions on names and on meta-infos. The result is a set of names of buffers.

For getting the meta-info of a specific buffer, the full meta-info can be requested to read. The Comm-server send a list of all meta-info; it can be empty too. If the requested name is not assigned to any buffer, the Comm-server rejects the request.

1. Buffer-query (regex buffername, regex meta-info)
2. Buffer-names (name1, name2, name3, ...) or Buffer-names ()
3. Buffer-read (nameX)
4. Buffer-meta-info (nameX, n1=v1, n2=v2, n3=v3, ...) or Buffer-meta-info (nameX)
5. Buffer-reject (nameX) There does no buffer exists with name=nameX

2.3.3 Process Communication Values

After the GBot has choosen a buffer, it can request the value of the buffer. It is possible to request the value immediately. It is possible to request the value on change.

Typically at the beginning the Gbot request the value immediately once. After this initial transfer of the value, the GBot has to request to be informed after next change of the value – repeately. If no request arrived at the Comm-server, the distribution of changed values will be stopped.

The transfer of the value will always be a broadcast. That means, the GBots are not differented by the Comm-server. If one or more then one request to be informed after change, the Comm-server send the next changed value to the Comm-client. The Comm-client distribute it to Gbots; arrange into complexData as requested.

1. Value-get-now (nameY)
2. Value-get-onChange (nameY)
3. Value-push (nameY, simplyData)
4. Value-reject (nameY)

2.3.4 Config Communication

The communication for configuration of buffers is not within the scope of this discussion. It must be done by Citem later.

3 Next Activities

The formal definition for the XML-telegrams – the XSD – has to be created; based on the XSD the appropriate Java-classes are generated automatically.

The communication for configuration of buffers has do be done.