Requirement definition

TCP-Listener KPU for PLC



Neutorstraße 13  
5020 Salzburg

AUSTRIA

Tel: +43 (662) 276198-11

Fax: +43 (662) 276198-98

Mail: [office@breanos.com](mailto:office@breanos.com)

|  |  |
| --- | --- |
| File | Anforderungsdefinition KPU.docx |
| Date | 12.04.2018 |

Table of contents

[1 Change history 5](#_Toc511312825)

[2 Requirement definition 6](#_Toc511312826)

[2.1 Introduction 6](#_Toc511312827)

[2.2 Scope 6](#_Toc511312828)

[2.2.1 Exclusion from scope 6](#_Toc511312829)

[2.3 Action plan 6](#_Toc511312830)

[2.4 Effort estimation 7](#_Toc511312831)

[2.4.1 Annotations 7](#_Toc511312832)

[2.5 Risk assessment 7](#_Toc511312833)

[2.5.1 Effort estimation incorrect 7](#_Toc511312834)

[2.5.1.1 Description 7](#_Toc511312835)

[2.5.1.2 Consequences 7](#_Toc511312836)

[2.5.1.3 Avoidance 7](#_Toc511312837)

[2.5.1.4 Probability 7](#_Toc511312838)

[2.6 Prerequisites 8](#_Toc511312839)

[2.6.1 Blackboard interfaces 8](#_Toc511312840)

[2.6.2 Hosting infrastructure 8](#_Toc511312841)

[2.7 Acceptance 8](#_Toc511312842)

# Change history

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Author | Description |
| 14.03.2018 | 1 | Krisch | Initial version |

# Requirement definition

## Introduction

The basic functional entity of any *Daipan* installation is the Knowledge Processing Unit, or KPU for short. The scope of this requirement definition is the first basic KPU as a functional proof of concept, implementing basic communication and processing functionality on the basis of existing requirements from previous projects.

The expected result will then be further improved to reflect extended and more complex functionality.

## Scope

The scope of this requirement is one fully functional KPU which is able to listen for incoming TCP connections, maintain a list of active connections, receive and send data over these sockets, handle the TCP socket life cycle, and process the known data input of a “Geely Status Update” (request and reply), provided by the existing PLC Simulator. The given data should be decoded, checked for plausibility, stored away to a Data Transfer Object, and forwarded to the blackboard system. An answer from the blackboard should be simulated and forwarded to the PLC Simulator accordingly.

The structure of the datagram received by the PLC simulator should be stored in XML, with an XSLT file describing possible (required and optional) properties and values / ranges, as well as request-reply-relationships. This XML should also include a separate node for header information. Reasonable properties have to be identified beforehand.

A strategy for resend detection and handling of resends / delayed or unavailable answers should be outlined, but an implementation thereof is not part of the scope.

Documentation before and after implementation in accordance with the *Vorgehensmodell* is part of the scope.

Reuse of existing code is encouraged where applicable. See therefor *PTS.Server,* as well as the Projects *Wuhan, Geely and Alfing* and the respective dependencies*.*

### Exclusion from scope

Excluded from the scope of this requirement is any further data processing inside the KPU or by different KPUs, forwarding of data to the client, additional data types available to the PLC Simulator, and displaying of information available to the blackboard.

The internal structure of the KPU is ad libitum, and no specific technology choice is part of this scope.

The necessary Data Mart can be simulated, or circumvented by writing directly to a file / DB.

## Action plan

* The KPU’s internal logic will be designed and documented in UML.
* The KPU’s external communication will be designed and documented in UML.
* The XML description will be designed in prose.
* Once these three design decisions have been made, they will be reviewed by the product owner.
* After being confirmed by the product owner, all requirements will be implemented.
* Every part will be tested during development.
* An integration test with the other framework parts will be done on April 27th.

## Effort estimation

The implementation effort is estimated as follows.

|  |  |
| --- | --- |
| Task | Hours |
| Design KPU | 12 |
| UML description | 8 |
| XML and XSLT | 6 |
| Interface definition | 6 |
| KPU Implementation (data logic) | 30 |
| Interface Implementation (TCP, Blackboard) | 18 |
| Internal testing | 12 |
| Documentation | 4 |
| Total | **96** |

### Annotations

This assumes pair programming, accounting for individual hours (so 12 hours estimate equals 2 men, 1 day).

## Risk assessment

### Effort estimation incorrect

#### Description

Seeing that this is the first KPU to be implemented, the effort estimation may be off by an impactful factor.

#### Consequences

Delays.

#### Avoidance

The Scrum model should support identifying this risk early on, and allowing for solution oriented support of the team. Additional manpower or support in specific tasks may be necessary.

#### Probability

High

## Prerequisites

The *Daipan* teams need to provide all necessary prerequisites to enable hosting, execution and external communication for the KPU.

### Blackboard interfaces

An interface to the blackboard (simulated or real) needs to be provided.

### Hosting infrastructure

Hosting infrastructure needs to be provided as required.

## Acceptance

Customer acceptance will encompass a test of all features described here, within the framework of the other *Daipan* components. Foremost it will include:

* Execution of KPU in designated environment,
* Communication with multiple PLC simulators in parallel,
* Connection handling,
* Protocol implementation “Geely Status Update”,
* Communication with the blackboard system,
* Data acquisition and storing,
* Test of adapted XML packet description.

Additional acceptance criteria will be agreed upon for each functional part.