## Circle Language Spec Plan, Commands Spec, Project Summary

*Author: JJ van Zon*

*Location: Oosterhout, The Netherlands*

*Date: June 21, 2008 – August 31, 2008*

### Goal

Work out the basic command topics.

Do not go deeply into parameters.

Leave out advanced command topics.

The main issue worked out in this project is commands being as free as an object.

### Super-project

This project used to be part of the project ‘Command As A Concept’, which proved to be too large, so it was split up into multiple projects.

### Time

June 28, 2008 – August 31, 2008

1 month and 3 days

**120** hours of work

### Means

One of the most difficult things was coming up with the creation behavior of calls.

Many times the article list was changed.

At one point I decided to isolate rules into separate articles and not repeat them everywhere.

That really helped.

It was much more work than I thought.

### Products

At the beginning of the project, the amount of articles to produce or adapt was 43.

Eventually the amount of articles produced or adapted was **110**.

The following was produced:

**106** articles were produced

**4** articles were adapted

*Commands, Basic article group*

version *2008-08-31 00 2.0x*

**106** articles: (**7** unfinished)

*- Commands*

*- Commands in a Diagram*

*- Commands Implementation* (not finished)

*- Execute Once*

*- Start & Stop*

*- Start & Stop in a Diagram*

*- Start & Stop Implementation* (not finished)

*- Executable Command*

*- Executable Command in a Diagram*

*- Inactive Command*

*- Inactive Command in a Diagram*

*- Changing Inactive to Executable*

*- Changing Inactive to Executable in a Diagram*

*- Command Definition*

*- Command Definition in a Diagram*

*- Command Call*

*- Command Call in a Diagram*

*- Command Reference*

*- Command Reference in a Diagram*

*- Clause*

*- Clause in a Diagram*

*- Inactive Clause*

*- Inactive Clause in a Diagram*

*- Active Clause*

*- Active Clause in a Diagram*

*- Command Anywhere*

*- Command Anywhere in a Diagram*

*- Sub-Command*

*- Sub-Commands in a Diagram*

*- Sub-Command-References*

*- Executables & Executions*

*- Executables & Executions in a Diagram*

*- Procedure*

*- Procedure in a Diagram*

*- Resolution When Not Allowed For Commands*

*- Parent Controls Its Sub-Executions*

*- Sub-Commands Are Never Referenced*

*- Sub-Commands Are Never Referenced in a Diagram*

*- Commands Never Contain Active Command References*

*- Commands Never Contain Active Command References in a Diagram*

*- Sub-Commands Not Manually Started*

*- Public Inactive Clauses*

*- Public Inactive Clauses in a Diagram*

*- Creation Behavior of Calls*

*- Creation Behavior of Calls in a Diagram*

*- Creation Behavior of Clauses*

*- Creation Behavior of Clauses in a Diagram*

*- Creation Behavior of ‘Inactive Calls’*

*- Creation Behavior of ‘Inactive Calls’ in a Diagram*

*- No Overhead Of Command Creation*

*- No Circular Command Creation*

*- No Circular Command Creation in a Diagram*

*- No Private Contents in a Call in a Definition*

*- No Private Contents in a Call in a Definition in a Diagram*

*- A Call in a Call Shows Privates When Running*

*- A Call in a Call Shows Privates When Running in a Diagram*

*- Active Command in Inactive Command*

*- Active Command in Inactive Command in a Diagram*

*- Reading & Writing Parameters*

*- Comparison to CPU-Like Calls* (not finished)

- Example Diagrams (folder)

*- Command Calls in a Command Definition in a Diagram*

*- Command Calls in a Command Call in a Diagram*

*- Command Calls in a Command Reference in a Diagram*

*- Parameters in a Command Definition in a Diagram*

*- Parameters in a Command Call in a Diagram*

*- Parameters in a Command Reference in a Diagram*

*- Private Objects in a Command Definition in a Diagram*

*- Private Objects in a Command Call in a Diagram*

*- Private Objects in a Command Reference in a Diagram*

*- Command References in a Command Definition in a Diagram*

*- Command References in a Command Call in a Diagram*

*- Command References in a Command Reference in a Diagram*

*- Clauses in Clauses in a Diagram*

*- Inactive Clauses in a Command Definition in a Diagram*

*- Inactive Clauses in a Command Call in a Diagram*

*- Inactive Clauses in a Command Reference in a Diagram*

*- Active Clauses in a Command Definition in a Diagram*

*- Active Clauses in a Command Call in a Diagram*

*- Active Clauses in a Command Reference in a Diagram*

*- Command Calls in an Inactive Clause in a Diagram*

*- Command Calls in an Active Clause in a Diagram*

*- Parameters in an Inactive Clause in a Diagram*

*- Parameters in an Active Clause in a Diagram*

*- Private Objects in an Inactive Clause in a Diagram*

*- Private Objects in an Active Clause in a Diagram*

*- Command References in an Inactive Clause in a Diagram*

*- Command References in an Active Clause in a Diagram*

*- Inactive Command Object Redirection*

*- Inactive Command Object Redirection in a Diagram*

*- Inactive Command Class Redirection*

*- Inactive Command Class Redirection in a Diagram*

*- Executable Command Class Redirection*

*- Executable Command Class Redirection in a Diagram*

*- Executable Command Object Redirection*

*- Executable Command Object Redirection in a Diagram*

*- Target Command Object*

*- Target Command Object in a Diagram*

*- Target Command Definition*

*- Target Command Definition in a Diagram*

*- Recursion*

*- Recursion in a Diagram*

*- Implementation of System Commands*

*- Command Object Referrers* (not finished)

*- Command Object Referrers in a Diagram* (not finished)

*- Command Definition Referrers* (not finished)

*- Command Definition Referrers in a Diagram* (not finished)

**4** articles were adapted:

- Target Class

- Target Class in a Diagram

- Normal Execution Order in a Diagram

- Basic Diagram Elements

use of object lines and class lines for commands is different now

Nothing, Public, Private.