## Circle Language Spec Plan | Parameters Spec | 2008-09 | Project Summary

### Goal

Worked out the Parameter articles, as part of working out Commands as a Concept.

### Super-Project

*Circle Language Spec | Command as a Concept Spec*

### Date & Time

September 4, 2008 – October 1, 2008

**4** weeks

**60** hours of work

### Background

Parameters can be considered the connection between commands and objects.

If a line merge can be used in bidirectional relations between classes, does that mean that a line merge between a command parameter’s class line and a class command’s definition line should be considered a bidirectional relation between a class and a command as well? Can that be viewed more like a relationship?

When a command definition gets an object with a class, this seems to create a relation between the command and the class definition. This means, that the class definition might also get a reference back to the command.

It looks like, when you turn a command into an executable object, without any additional effort automatically seems establishes the interchangeability between class commands and command parameters. That was an unexpected turn. If two seemingly independent concepts confirm each other, maybe it's on the right track.

### Products

At the beginning of the project, the amount of articles to produce or adapt was 47.

Eventually the amount of articles produced or adapted was **27**.

The amount of articles shrunk, but the amount of *work* did not.

The following was produced:

**27** articles

*Parameters articles*

version *2008-10-01 00 1.0*

**24** articles

*- Relations Between Commands & Objects*

*- Relations Between Commands & Objects in a Diagram*

*- Access Controlling System Aspects*

*- Access Controlling System Aspects in a Diagram*

*- Simplified Access Control Expression*

*- Simplified Access Control Expression in Text Code*

*- Simplified Access Control Expression in a Diagram*

*- Joint Display of Access Connectors & Object Relations*

*- Imported Parameter Concepts*

*- Required & Optional*

*- Required & Optional in a Diagram*

*- Required & Optional in a Text Code*

*- Variable Amount of Parameters*

*- Variable Amount of Parameters in a Diagram*

*- Variable Amount of Parameters in Text Code*

*- Return Values*

*- Return Values in a Diagram*

*- Parameter Order*

*- Parameter Order in a Diagram*

*- Parameter Order in Text Code*

*- Miscellaneous Parameter Topics*

*- Parameters & Arguments*

*- Parameters & Arguments in a Diagram*

*- Parameters For The Add Command*

*System Objects articles*

3 articles

*- Parameters For Objects*

*- Parameters For Objects in a Diagram*

*- Parameters For Objects in Text Code*

The original list was **47** articles, which included:

- Value topics

> Moved to Assignment

- Old parameter passing type subdivision

that used the terms input and output.

> Moved to Advanced Command Topics.

- Imported and misc parameter topics

> Also there in the eventual list.

The original list did not include the idea of Joint Display of Access Connectors & Object Relations.

### Means

The idea was set aside, that parameter notations might have discrepancies with automatic containment.

The idea was set aside, that the notation of automatic containment for relations was not much worked out yet.

Some key phrases out of the super-project description:

- Try to form a clear idea in your head.

- It should be a good idea to look at the broader view too.

- Maybe put more effort into it when changing the approach to the work.

- Maybe do not go for producing the articles as fast as possible.

- Consider to not dismiss an idea, when it might not be easy to work out.

### Reflection

This reflection attempts to sum up which turns were taken in the project.

At first there were the following pieces of material to organize and turn into a single Parameter documentation section:

- Commands & Classes Loosely coupled

A longer story about the concept of commands & classes loosely coupled and interface reliability and the internet.

- Older parameter type subdivision

The idea at first was that it would be converted to the eventual article list, but that was later not done that way.

Value In

Value Out

Value Thru

Reference In

Reference Out

Reference Thru

Object Out

- Older, simpler descriptions of the terms parameters and arguments:

Parameters

Parameters in a Diagram

Arguments

Arguments in a Diagram

- A possible parameter article list:

This eventually became the imported and misc topics.

First the material was read over and comment was dropped into it.

The texts of the older parameter type subdivision was rephrased and reorganized. At first the plan was, that this was to become the eventual article list, but eventually it didn’t.

Then the Commands Basics documentation section was read over in search of ideas about more parameter passing types. The parameter passing types that were planned to be added, had to do with the class aspect of a parameter and that parameters can also be commands. The spectrum of possibilities seemed clearly formed then.

Then an attempt was made to convert the idea of Commands & Classes Loosely Coupled into Relations Between Commands & Objects.

Quite early it became confusing. When looking at parameters as relations between commands & objects, then object parameters usually point outwards. It seemed hard to match with the reference situations thought of earlier: pointing *to* a parameter and let the parameter point to something.

After some brainstorming the whole idea seemed to land on its feet again. The display of relational structure between commands and objects could be displayed separate from the parameter passings: access connectors of the ways you can access the parameter. By separating those two things, the system seemed to be able to function again. Most of it seemed to go smoothly then, but after working out some topics smoothly, something more difficult came up: coming up with simplified parameter access control literals, which eventually kind of failed and existing unfinished ideas about simplified parameter access control literals were just described instead.

Next, most of the material from Commands & Classes Loosely Coupled was moved to the documentation sections Interfaces and Public & Private.

Next, it was not so easy to cross out the articles about old parameter passing types. At one point a good plan seemed to be followed, but soon after starting, the plan seemed to fall apart again. The thing was: much of the text mentioned input and output and no clear definition of the terms input and output was found yet. It seemed unclear whether output was what’s read on the outside, or output was what’s written on the inside, and for a while it seemed it was both, but later it seemed it was not.

Eventually it was decided to move most of the sections about the old parameter passing types to the Advanced Command Topics, and decided the whole definition of the terms input and output might be determined in the project of working out the Advanced Command Topics, because those already go deeper into the concept of input and output.

Sections of the old parameter passing types, that were about value transfer, were reformulated and moved to the Assignment topics.

Then the remaining brainstorm texts in the project description were crossed out. Lastly, the even older parameter articles were crossed out.

Then there was a phase that was less difficult: documenting the imported and miscellaneous parameter concepts. That was actually one of the easier things. It was possible to finish almost all of that in one evening while being very, very tired.

Lastly, a good start was made at the miscellaneous topic Parameters For Objects, but it was moved to the documentation section System Objects.

The last phase in the project, was wrapping up the project.

In this project things were stressful at work. It influenced the amount of time that could be put in it in spare time, and it influenced clarity in the head.

It was quite a difficult project. Needing to dig deep to try and come up with the right ideas.

### Project Steps

#### Preparation Work "Parameters"

(0h)

- Considered the general approach, described in the super-project *Circle Language Spec.*

- Commands & classes loosely coupled was moved to the parameter articles.

- Recorded the amount of articles, that was planned to be written at the beginning of the project.

#### Reading over Existing Texts

(9h)

- Reading over the material and dropped comment into it

except for Commands & Classes Loosely Coupled.

- Reading over Commands & Classes Loosely Coupled.

- Reading it over.

- Verbal brainstorming.

- It would later be used as a cross out list.

#### Prepared Texts

- Adapted parameter texts

- The Input, Output, Throughput, Downput introduction seemed to consist of multiple disparate details, that could be moved to more specific articles, because the ideas were later repeated.

- Removed the separation between direct and indirect value transfer.

- The 'reference situations' are not mentioned, before the parameter passing summary.

> Renamed *Parameter Passing Summary* to *Three Parameter Passing Elements.*

> Came up to Object Out in a Diagram.

- ~ Remaining ideas might only be worked out later.

- Extended it with more types of parameter passing:

- Based on Command Basics.

- Gathered up issues.

- The issues:

- The issues that have to do with:

- Parameters that are commands

- Classes

- Specific issues:

- Parameters that are command references.

- Parameters that are output commands.

- Existing command

- New command

- The class of a parameter.

- The class of a parameter, that is a reference.

- Public active command references inside commands.

- Proposed rule: Sub-commands are never referenced / sub-commands are never output objects.

- Issues were partly written out.

- But more issues were found.

- Extended with more types of parameter passing:

- There were more possibilities.

- The spectrum of possibilities was in part worked out on paper.

- But now there is a tendency to want to start over at the start of the story again.

- Rewriting Commands & Classes Loosely Coupled.

- A new name could be ‘Relations Between Commands & Objects’.

- It might not be clear yet exactly how the article set would look or how the parameter types will be like.

- First wrote about relations between commands & objects.

- Then wrote about all the different reference situations parameters can have.

- Then wrote about how you can access control each situation.

- And why you might access control each situation.

(object algebra, deduction, concurrency resolution, control by user)

(only command Input/Output might lead to that, actually… anything about sub-object access does not seem to be covered. That might be covered in Advanced Command Topics.)

- Tried to cover the use of each reference situation.

- Then wrote about access control literals.

#### Changed Approach

The approach was changed, because starting at Relations Between Commands & Objects the idea about reference situations changed. After brainstorming, a direction was found again.

- Preparation:

- Brainstormed about different approach.

- Isolated an unfinished version of article group.

- Articles and processing:

- Relations Between Commands & Objects

- Relations Between Commands & Objects in a Diagram

- Object relations

- Class relations

- Access Controlling System Aspects

- Access Controlling System Aspects in a Diagram

- Simplified Access Control Expression

- Simplified Access Control Expression in Text Code

- Simplified Access Control in a Diagram

> It would be nice to see the different kinds access connectors with the new textual literals.

- The Data aspect

> Indirect value access from the inside was not mentioned yet.

> It could be seen as another system aspect.

- Joint Display of Access Connectors & Object Relations

#### Crossed Out Old Material

- Crossed out ideas in project document.

- Crossed out Commands & Classes Loosely Coupled.

- Because the idea of Commands & Classes Loosely Coupled could be based on 'relations between commands and objects'.

- Texts have been moved to other article groups:

- Interfaces

- Public & Private

- Crossed out old parameter passing type articles.

- Reading over brainstorm sections near the end, that were made before changing the approach. Probably delete most of those texts.

> Two pieces of the text were usable in other article groups.

> And two sections could become an article in the Parameters article group.

- Setting aside assignment topics.

- Topics:

- Clone

- Value Direction

- Indirect Value Transmission

- Moved old parameter passing types to Advanced Command Topics.

Because they talk about In, Out and Thru. It was estimated that the definition of those terms might be harder to do, and might have to do with the Advanced Command Topics.

- Added it as a possible goal of the project Work Out Advanced Command Topics.

- Putting a sort of disclaimer in the articles for Simplified Access Control Expression about the use of the terms In Out and Thru.

- Putting in the plan to adapt the articles Simplified Access Control Expression, being more reserved about the terms In, Out and Thru.

- Assignment related

- Wrote Clone section for the assignment articles.

- Moved to the assignment articles:

- Clone

- Value Direction

- Moved to the Advanced Command Topics:

- Indirect Value Transmission

- Wrote something about cloning access connectors.

- Registered possible future project Update Assignment Articles.

- Crossed out remaining ideas in the older parameters document.

- Changed the articles Simplified Access Control Expression, being more reserved about the terms In, Out and Thru.

- Wrapped up brainstorm texts in this Project document.

- Crossed out older parameter articles.

- The older articles contain mostly information about Commands & Objects Loosely Coupled, and were integrated into the story.

#### Misc Articles

- Mainly texts of miscellaneous articles were considered as an option to write.

- Wrote them: (**19**)

- Imported Parameter Concepts

- Required & Optional

- Required & Optional in a Diagram

- Required & Optional in Text Code

- Variable Amount of Arguments

- Variable Amount of Arguments in a Diagram

- Variable Amount of Arguments in Text Code

- Return Values

- Return Values in a Diagram

- Parameter Order

- Parameter Order in a Diagram

- Parameter Order in Text Code

- Miscellaneous Parameter Topics

- Parameters & Arguments

- Parameters & Arguments in a Diagram

- Parameters for the Add Command

(Preliminarily finished and moved to the System Objects documentation section:)

- Parameters For Objects

- Parameters For Objects in a Diagram

- Parameters For Object in Text Code

- Reading them over

#### Processed Left-Over Ideas

#### Wrap up

- Gave articles a parameters-specific article name.

- Setting up article list.

- Isolated version.

- Turned preparation work 'Computer Language General' into a separate project.

- Made the next project also an 'Organize Computer Language Ideas' project.

- Wrapped up project description.

- Reflected on how this project progressed.