# Circle Language Spec Plan | Planning + Black Box Spec Part A | 2009-06 | Cycle 3: Redo System Objects & Assignment A

## Purpose of This Document

This document is a checklist for software development-related work, for a good approach and to not forget anything. It is not to be read over literally. Later it serves as a reference for looking up how exactly something was done.

## Reflection

This phase is where it went wrong. Work spiraled out of control.

## Rough Plan

Black boxing has a lot to do with assignment and system aspects. The black boxing notations need to match with the assignment notation and the system objects notation, so you have to make the notation in the Assignment articles and in the System Objects articles ok.

- Update Assignment Articles

- Get the notation straight

- Update System Objects Articles

- Get the notation straight

Also, parts out of the Black Boxing documentation need to be moved to the Assignment and System Objects articles, where they belong.

In the end, the Assignment articles were integrated into the System Objects article group, because it was easier to explain those together.

## Approach

I introduce things in wrongly composed chunks in the System Objects documentation.

- What I am missing is immediate overview of System Objects (e.g. Related Item, List Item), immediate overview of System Aspects, and then the definition of a system command.

- It starts out with a very difficult story

- Way too many details seem to be mentioned.

- Perhaps you should first explain the System Objects

- And only then their system aspects

- And perhaps even separately the system commands

- Base it more on a story about System Aspects.

This should be the new order:

- All the system objects in general and how they relate to each other

- All the system aspects in general

- The system commands in general and per aspect

- The system interface in general

- All the system objects with the system interface notation

> Without system commands in it?

- System commands in the system interface notation

## Steps

### Additions

X Object-Bound and Reference-Bound

X Moet naar daar verplaatst worden.

/ En de rest aanpassen aan onderscheid Object-Bound en Reference-Bound

-> Hoefde niet. Alleen Value is object-bound in het bestaande materiaal.

X System command call notation:

X Calls without argument / counterpart

X Calls with an argument / counterpart

(zonder counterpart is het eigenlijk niet compleet.)

(laat het argument niet expliciet zien in de call:

het argument is aan de andere kant van de lijn)

X Explicit Get with implicit Set

X Assignment

X Integrate assignment into it.

-X Split up into multiple articles.

~X Assignment:

X Assignment notatie moet iets vereenvoudigd

Hele ding in een bepaald line type, inclusief direction mark en assignment call symbool

X Reference assignment hoeft niet per se een pijl,

tenzij het een assignment is van een reference op hetzelfde niveau.

maar in alle andere gevallen is de source van de assignment inward.

> Eigenlijk wil ik dat wel, omdat het een heel speciaal effect heeft, pointer assignment.

X Access connector notation:

X Then the system command connectors (get, set and use do not show a command symbol)

X Represents a potential call

X Which also makes it represent the system command definition, that can be called upon

X System Commands zou op zich al access connectors kunnen behandelen,

maar dan zonder Private en Friend, dus zonder black boxing.

X Access connectors zijn ook potentiele calls naar system commands

die moeten ieder lijken op het resultaat of anders op de call.

X The explicit notations:

X Philosophy: the explicit, non-simplified system command call notation should be worked out

last with a clarification for why it gets a more basic notation.

X 1) System Command Call with Argument / Explicit Get, Implicit Set

Add the cross-aspect things

X 2) Explicit Get, Set and Use (both counterparts explicit)

X 3) Explicit display of Get and Set arguments

X One article:

X 4) Completely explicit display: showing the system interface and the exactly

correct line types.

X 5) Explicit display of assignment

(the name of the assignment command is for instance 'o='. Completely explicit display also means displaying that name.)

X Maybe integrate the last two in the same article, because they look so similar.

X In the article put in a comparison to the substituted notation of assignment.

X Connections

(a new addition)

X You would also want Connections to have an article in the system objects article group.

### Reformations

- Topic list:

X System Objects

X The Object

X The Reference

X Related Object

X Related Item

X Related List

X Item

X Related List Item

X Symbol

X Related Items & Related Lists Collections

X System Aspects

X Reference

X Object

X Class

X Value

X Clone

X Name

X Existence

X Data

X Execute

X List

X Interface

X System Commands

X For Reference Aspect

X For Object Aspect

X For Class Aspect

X The Extra Commands, Overloads & Delegation

X For Value Aspect

X For Clone Aspect

X For Name Aspect

X For Data Aspect

X For Execute Aspect

X For Existence Aspect

X For Add & Remove Aspect

X System Interfaces

X General System Interface Notation

X For Object

X For Related Item

X For Related List

X For Related List Item

- Rules:

- Small plan:

X Try to summarize the rules

X Add stuff you forgot to mention

X Read over existing material to see where the rules are mentioned

- And isolate the rules in separate articles, instead of repeating them.

X Forgot to mention:

X I forgot to mention the optional Class argument in the System Commands for the Existence Aspect.

X And perhaps also the arguments in the Add command in the System Commands for the List Aspect.

X Rules repeated:

X Use-Commands Get Another Aspect

X Aspect-In-A-Triangle design choice

X 24. System Interfaces of Objects & References

X Final choice:

X Reference system interface by default,

within which the Related Object is visible, with its system interface showing.

X Practical other option:

/ Display both object-related aspects and reference-related aspects in the same system interface.

> A symbol can have multiple system objects, so multiple system interfaces.

> For instance: one symbol can have an OBJECT system object and a related item system object.

/ Impractical options:

/ Show Object system interface in imaginary reference.

Show Reference system interface everywhere else.

But that will make it hard to for instance get a Value through a reference.

/ Always show Object system interface, and through it its related items.

But then you can not directly work with related items,

you would have to go through the system interface of its parent object.

> Oh, and the object system object will have a related lists and related items collections.

X Preliminariness

X How the system interface looks is totally dependent on how the code base

is implemented. If you make different design choices about the code base,

organizing members differently, the system interface will also look different,

because the system interface is an exact representation of the public members of the system object.

Not covered:

/ Show aspect as property

X Detail:

X Use more lidwoorden in titles for commands for aspects

X Read over

X After you put the existing newer topics:

X Assignments

X Connectors & Connections

X System Command Calls

X Explicit Notations

(not necessarily in that order)

(you can simplify them)

> At this point you have a fairly well finished up chapter,  
except for a lot of cross-out material.

### Cross-Out + Additions

X Cross out old articles

X Remaining topics:

X System interface command call

~X System Command Extension

> Preliminarily Finished

~X Parameters For Objects

> Leave this way

X Ancestry Terms

/ Transform Method To Object (cross out)

> Content redistributed

X Cross out Ideas (+/- 60 pages)

> I was worried about a lot of extra aspects, to control other system object members, and a lot of extra Get purposes, but there are not that many left. Just a couple.

### Work Discontinued

There were other cross-outs, but they are postponed.

At this point work had spiraled out of control.

This project is wrapped up and a replanning is done after which the work continues as a new project.