## Circle Language Spec Plan, 2008-07 Assignment Spec, Project Summary

*Author: JJ van Zon*

*Location: Oosterhout, The Netherlands*

*Date: August 3, 2008*

### Goal

Work out the Assignment article group, required to work out Commands as a Concept.

### 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.

### Date & Time

July 24, 2008 – July 28, 2008

5 days

12 hours of work

### Products

The following was produced:

**32** articles were produced

**2** articles were adapted

**1** other item of work was done

*Assignment article group*

version *2008-07-28 00 2.0*

**30** articles:

*Assignment*

*Assignment in a Diagram*

*Assignment in Text Code* (=) (not finished)

*Value Assignment*

*Value Assignment in a Diagram*

*Value Assignment in Text Code*  (v=) (not finished)

*Object Assignment*

*Object Assignment in a Diagram*

*Object Assignment in Text Code* (o=) (not finished)

*Class Assignment*

*Class Assignment in a Diagram*

*Class Assignment in Text Code* (c=) (not finished)

*Interface Assignment*

*Interface Assignment in a Diagram*

*Interface Assignment in Text Code* (i=) (not finished)

*Object Reference Assignment*

*Object Reference Assignment in a Diagram*

*Object Reference Assignment in Text Code* (=🡪) (not finished)

*Object Reference Object Assignment*

*Object Reference Object Assignment in a Diagram*

*Object Reference Object Assignment in Text Code* (o=🡪) (not finished)

*Object Reference Class Assignment*

*Object Reference Class Assignment in a Diagram*

*Object Reference Class Assignment in Text Code* (c=🡪) (not finished)

*Object Reference Interface Assignment*

*Object Reference Interface Assignment in a Diagram*

*Object Reference Interface Assignment in Text Code* (i=🡪) (not finished)

*Alternative Assignments*

*Alternative Assignments in a Diagram*

*Alternative Assignments in Text Code* (not finished)

Object articles: **(2)**

*Nothing*

*Nothing in a Diagram*

(only added to the development version of the Objects article group)

Adapted articles: **(2)**

*Basic Diagram Elements*

(add the wavy line)

*Sub-Object*

The definition of sub-object is not right anymore,

because I say in the Object Reference article,

that object references are no longer called sub-objects.

Other work: **(1)**

Isolate Storage Principles

(has nothing to do with the project, but I want to do that.)

### Project steps

The notation for assignment is made simpler.

Command assignments the same as object assignments.

Extended with object reference assignments and alternative assignments.

Details:

- Old set of articles organization:

- Assignment

- Assignment in a Diagram

/ Value Assignment

- Value Assignment in a Diagram

/ Object Assignment

- Object Assignment in a Diagram

- Command Definition Assignment

- Command Definition Assignment in a Diagram

- Class Assignment

- Class Assignment in a Diagram

- Interface Assignment

- Interface Assignment in a Diagram

- Command Interface Assignment

- Command Interface Assignment in a Diagram

The set of articles will change

At first there was only direct assignment of:

- Value

- Object

- Class

- Interface

- Command Definition

- Command Interface

Commands will become a concept, automatically making Command Definition the same as Class, and Command Interface the same as Interface. That leaves us:

- Value

- Object

- Class

- Interface

You do need to mention it in the articles when an assignment is also synonym for a command assignment.

Pointer assignment adds more types of assignment. Object, Class and Interface assignment each get two types of assignment:

- Object assignment:

Assigning the object as the target

- Object reference assignment:

Assinging the related object as the target

Any alternative form of assignment can be accomplished by calling system commands and assignment commands.

This creates a different set of articles:

- Assignment =

- Assignment in a Diagram

- Assignment in Text Code (not finished)

- Value Assignment v=

- Value Assignment in a Diagram

- Value Assignment in Text Code (not finished)

- Object Assignment o=

- Object Assignment in a Diagram

- Object Assignment in Text Code (not finished)

- Class Assignment c=

- Class Assignment in a Diagram

- Class Assignment in Text Code (not finished)

- Interface Assignment i=

- Interface Assignment in a Diagram

- Interface Assignment in Text Code (not finished)

- Object Reference Assignment =🡪

- Object Reference Assignment in a Diagram

- Object Reference Assignment in Text Code (not finished)

- Object Reference Object Assignment

- Object Reference Object Assignment in a Diagram

- Object Reference Object Assignment in Text Code o=🡪 (not finished)

- Object Reference Class Assignment

- Object Reference Class Assignment in a Diagram

- Object Reference Class Assignment in Text Code c=🡪 (not finished)

- Object Reference Interface Assignment

- Object Reference Interface Assignment in a Diagram

- Object Reference Interface Assignment in Text Code i=🡪 (not finished)

- Alternative Assignments

- Alternative Assignments in a Diagram

- Alternative Assignments in Text Code (not finished)

You have to take the following details into consideration:

> You can assign a class to a sub-object, but you can also assign a class to an object. When and how do you do which?

> I have only described assigning it to a sub-object.

> Perhaps you first assign a class to a sub-object and then create the sub-object . You can’t change the class of an object after it is created.  
You *have to* mention this in class assignment.

> You always access an object through an object reference, and an object reference is always part of the object reference’s parent. So you are always accessing an object through another object.

> What if you want to assign a pointer to a class as the class of a symbol, instead of assigning a class to a symbol?

> I guess normally you’ll just assign the class of a symbol to another symbol. If that class is a pointer to a pointer to a class, you’re assigning the pointer to the pointer to the other symbol. Assigning the target class may need to be another type of assignment.

> In that case, you need to leave out the term target in the normal assignments, and add a new set of possible assignments: target object assignment, target class assignment, etcetera.

- Alternative assignments, not target assignments

What if you want to assign the Target Class as the class target of an object reference. I could imagine: c=c🡪🡪

You could do the same for interfaces.

But you could do the following anyway:

A c=🡪 B . Target Class

The diagram is worked out on paper.

Object reference assignment is mandatory.

But target object reference assignment is not. Target object reference assignment is less common, and possible in an alternative way. Target object reference assignment also leads to thinking about assigning other types of targets and it would clother the language with a bunch of alternative notations for things that are less common, for which a general notation can be used.

> Commands and objects more the same

Class and definition assignment are the same in that case,

and then there is only one type of interface assignment.

> Calls are created briefly, but definitions are created permanently.

- How does that influence the different types of assignment?

- It means definitions could get object lines anyway, and calls can not.

- Articles to change:

- Object assignment

- Object reference object assignment

- Class assignment

(rephrase small parts)

- Object reference class assignment

(rephrase small parts)

- Incorporate:

To make direct conversion between a command and an object

possible without any loss of structure, a call must be

able to have an object line

In that case the rule is: when a call symbol has an object line,

the object line behaves as a class line.

But it does require a warning to be generated.

- Explain unsimplified notations of assignment

I should display the notation of an assignment calling upon system commands.

I didn’t know how to do it before, but now most is known about the system interface, I do know how to express it.

Object assignment:

(Picture -1)

B . Object . Set ( A . Object . Get )

(Picture 0)

B . Object . Set ( A . Object . Get . a Value )

(Simplified) (because in an assignment you do not need to show the words Get and Set, but s general assignment does not exist)

B . Object = A . Object

(Picture 2 ½)

o= ( B , A )

B o= A

(Picture 3)

B o= A

Object reference object assignment:

B . Other Related Item . Set ( A . This )

B . Other Related Item . Set = A . This

B o=🡪 A

- Articles to adapt:

- Assignment

- Assignment in a Diagram

- Assignment in Text Code

- Object Reference Assignment

- Object Reference Assignment in a Diagram

- Object Reference Assignment in Text Code

- Wrap it up with more references to other articles:

- In the main article refer to all the sub-articles.

- In all the sub-articles refer to the main article.

- Maybe references to the articles target objects, target classes and target interfaces

> Only referenced target objects in the article Alternative Assignments.

/ In Target Object Reference Assignment and Object Reference Assignment refer to the article Automatic Containment for an explanation on imaginary reference lines.

- Put all references to other articles in the see also list

- Class and interface assignment:

If an object does not have a class,

then the object itself should be assigned as a class.

If an object does not have an interface,

then the object itself should be assigned as an interface