|  |
| --- |
| Circle Language | Broader View |

## Black Boxes Broader View Ideas

### Side-Issues

In previous projects the main articles about black boxing were finished, but other subjects were left unfinished. This article contains the unfinished material of some of these topics.

This article contains texts and loose ideas about the following topics:

*Black Boxing and User Access Control*

Covers how simple black boxing could be made parallel to user access control.

*Programmers and Users*

Addresses mainly the issue why when something is private, programmers get to change everything anyway, no matter how private or inaccessible things are. And why do users not get that privilege and what stops them from getting that privilege.

As said, the texts about these subjects are not finished yet.

#### Black Boxing and User Access Control

Access control will be fully managed by the access control module, even for access controlling an object's members, making the members only accessible to the direct container of the member in order to hide complexity from the outside, or to to protect the object's data.

But that form of access control is so important in programming software, that it is introduced earlier as the concept of Public & Private, before access control is worked out as a complete solution to security.

Maybe access controllers are a step into the direction of security. Maybe security should be a kind of advanced access control.

Interesting idea for user access control: a constant in a program is a variable that can be changed by a programmer, but not by a user and also not by the program itself.

-----

Public & Private,

± 2008-09

This project should be the working out of the concept of Public & Private, even before it is integrated into the complete Access Control concept.

Public & Private need a preliminary notation, and preliminary explanation, so that the concept can be used in explanations of coding concepts whereever required.

JJ

-----

Private & Public,

2008-06-10

Access Controlling System Aspects:

Save for user access control, because this is about different roles:

You can have friends for reading, writing

or changing (add and delete symbols.)

JJ

#### Programmers and Users

black boxes = user access control?

privates visible in friends, and in definitions when you have user access to it, so you are a programmer, not a user.

Something like that…

-----

Private & Public,

2008-06-10

Editing definition:

> 2009-06-30 Some of the ideas here are wrong, but the general idea should be thought of about editing the definition.

Friend access is different from access to private contents

only when editing the definition.

> 2009-06-26: No. this is user access control. Some users are authors of a definition. Some are not. Do not confuse it with private/public concepts.

It's the difference between friend for reference,

and friend for change.

> 2009-06-26: Meh, see last comments.

If you are a friend for change (not for reference) for private contents, you will only see the private contents if you are at logical target the definition.

JJ

-----

Set and Copy are used to set up lines. Lines are set up at design time, but can also be changed at run time. Sometimes you want to set a line at design time without being able to change it at run time. Therefore, Set and Copy must be able to have different access controllers at design time and run time. Just like the Get purposes can have different access controllers, it’s possible to have different access controllers for Design Time Set and Run Time Set and for Design Time Copy and Run Time Copy. Something is a constant if it can be Design Time Set, but not Run Time Set.

Object symbol system procedures and purposes:

Symbol

Get : Copy (Run Time or Design Time)

Object

Get : Copy (Run Time or Design Time) or Redirection

Set : Run Time or Design Time

Type

Get : Copy (Run Time or Design Time) or Redirection

Set : Run Time or Design Time

Interface

Get : Copy (Run Time or Design Time) or Redirection

Set : Run Time or Design Time

State

Get : Copy (Run Time or Design Time)

Set : Run Time or Design Time

Target Object

Get : Copy (Run Time or Design Time) or Access

Target Type

Get : Copy (Run Time or Design Time) or Access

Target Interface

Get : Copy (Run Time or Design Time)

Procedure symbol system procedures and purposes:

Symbol

Get : Copy (Run Time or Design Time)

Call

Get : Copy (Run Time or Design Time)

Set : Run Time or Design Time

Reference

Get : Copy (Run Time or Design Time) or Redirect

Set : Run Time or Design Time

Target Procedure:

Get : Copy (Run Time or Design Time) or Access