

Annotations 10/03/2009

For a component, we must have informations like his name, sub-components, interfaces provided and required. We must also make link between the major component interfaces and sub-components interfaces.

We will make 3 annotations, in order to simplify the comprehension of them.

```
@component(name, {list of sub-components})
```

```
@provided({list of name, {list of links}})
```

```
@required({list of name, {list of links}})
```

This way, if a component doesn't have required or provided interface, we do not need to specify an annotation. Otherwise if we would have made the choice of giving these informations in the annotation `@component`, we would have need to make test on the number of parameters. With this method, we only need to test if the annotation exists. In the case where we don't have sub-component we simply will specify a blank.

With this architecture, the research of the structure of the program is easy to understand, we could with the help of a program (which can analyze annotations), draw automatically each of the diagrams of the program. Basically only the structure not methods.

We should certainly extend the list of sub-components of the annotation `@component` with every sub-classes in order to specify the role and the area of action of a defined class. A class of a sub component should not be used by his father, and not by his child.