Unit Management

Task

Units are program technical units, which are dynamically integrated into the CBA-Framework. The units implement methods, that realize the functionality. They present besides configuration and descriptive specification the third way for implementation - the program technical implementation. The CBA-Units are to a large extent parameterizable, so that the implementation of reusable program technical unit is not only possible, but also supported in a very good way.

The unit concept is a core piece of CBA and was designed on the basis of a proven Service-Design-Pattern for service-oriented applications. Its program technical implementation was with the help of the CBA-Framework a consistent continuation of this aspect. The goal is to create small, methodically clean developable and efficiently feasible units, that work together with uniform principles. Especially aspects of loose coupling between porgramm technical units played an important roll. The Units themselves are parameterised and the information flow between them is configurated.

The CBA-Unit-Concept builds on a 3-Layer-Concept. On every layer exists a kind of programm technical units with defined interfaces

- Data Access Layer
- Business Logic Layer
- User Interface Layer

The Unit Management knows the configurated units and instantiates the Units dynamically at the time they are called for the first time.

PICTURE UNIT MISSING

The CBA-Framework offers for every layer layer-specific acces functions, which determine the right Unit and call the right function. The CBA-Framework and its predecessor product ADI, that works with the same principles, have proven themselves in long-standing work and many products. It is stable, tried and tested, but still enables every kind of expansion. This openness enables the implementation of diverse applications on the basis of the CBA-Framework.

The strong decoupling of programm technical units on the one hand forces a clear software structure, but on the other hand supports the software development process very good methodically. Through the specification of programm technical Design-Pattern the code is easily understandable. The units' different methods have well defined tasks, so that a easy and explicit description of the program logic is already realizable in the design. The code can also often be implemented without a formal description. This feature is intended and enhances agile methods. Alternatively has the description in form of pseudo code also proven to be beneficial in the program technical frames. E.g. For bigger tasks models can be created with UMl and implemented in a seperate step.

Architecture

The CBA-Framework knows three Unit types:

- DAU - Data Access Unit
- BLU - Business Logic Unit
- UIU - User Interface Unit

In principle the CBA-Unit-Management can also manage more Unit types. The picture gives a overview over the structure of interfaces and classes. When implementing new Unit types the architecture principles are to comply with.

PICTURE MISSING

Every Unit type realises a type-specific interface, which is derived from the interface **ICbaUnit**. **ICbaUnit** defines the functions, that are necessary for the dynamic management of the units. The interface or implementations are dervied form the interface **ICbaUnit** or the provided base implementation **CbaUnit**. For that CBA defines the interface or the base implementation

- Data Access Units: **ICbaDau** or **DauBasic**
- Business Logic Unit: **ICbaBlu** or **BluBasic**
- User Interface Unit: **ICbaUiu** or **UiuBasic**

Ever Unit is implemented via one class. The Unit-Management instantiates objects of the configurated class. In the configuration of the units the DLL, which containes the **class**, is put under **Assembly**. The statements about the Assemblies take place without the file expansion *.DLL. The class is given with the namespace.

In the CBA scope of delivery is a set of standard implementations present. The from the system offered Units are in each case derived from a type-specific base class.

Configuration

New Units can be implemented and added to CBA in the following way:

- Implementation of the desired functions by deriving from the base class
- Integration of a new data set into the Unit-Management under the menupoint configuration - units under specification of all required configuration parameter
- Upload of the Assembly and the documentation

For the duration the Unit-Management instantiates the objects of the configurated class as a Unit and dynamically integrates them into the information and control flow.

Functions

For the Unit-Management the following functions of the CBA-Framework are available:

- **getUnit:**instantiates the Unit of the given type and the given ID
- **getDau:**determines or instantiates the configurated DAU for the table . Every DAU is only instantiated once, even if it is configurated for multiple tables.
- **getBlu:**determines or instantiates the configurated BLU for the table . Every BLUis instantiated

once for every table
- **getUiu:**determines or instantiates the configurated UIU for the user interface. Every UIU is instantiated once for ever user interface

getUnit is only used for internal needs, but also offer the possibility to manage other Unit types with the CBA-Unit-Management. For this purpose it makes sense to implement similiar functions like getDau, getBlu or getUiu, that can be accessed from the outside.

The functions of the Unit-Management are mainly called from the CBA-Framework. But for the expansion of the Framework they are uselful.