PITSS.CON Application Engineering [AE]
For Oracle Forms and Reports Application

Bring your Forms Application to a State of the Art

Software Architecture

PITSS.CON 8.0.0

White Paper, November 2009

1.   Objective

The present document aims to present the "Application Engineering" AE concept based on the PITSS.CON tool in very simple terms. This concept has been devised to transfer Oracle Forms applications into modern, future-oriented architectures.

2.   Approach

In order to transfer proprietary software, which is mostly Oracle Forms applications, into a modern technology, you need to understand the application, functions and processes. As Forms was hailed as the programming language of the 4th generation decades ago, combining the graphic interface, process and associated logic in only one source, it must now be disassembled into these components for the technological transformation. Manually, this is hardly possible, if so, it can only be done with a high level of Invest. PITSS.CON offers a technical (automated) approach.

This approach is based on the extraordinarily powerful development environment of PITSS.CON, which specifically accompanies Oracle Forms applications along their "life cycle".

The core of PITSS.CON is a repository in which the Forms applications can be analysed according to their most elementary components, disassembled, documented, monitored, manipulated and developed further. Based on these capabilities, Forms programs are disassembled into their components GUI / processes / logic, in order to then be transferred to the new technologies.

3.   PITSS.CON Module "Application Re-Engineering" at a glance

The code based on PL/SQL in a Forms application is identified, analysed, extracted and transferred to database packages in a targeted manner. A developer is guided when doing so, so that he or she can transfer the previous Form logic, consisting of libraries and database objects, into a neat database structure.

PITSS.CON follows the approach of creating a database structure that is based on so-called layers. A layer performs a clear functional task and adopts the associated logic.

The basic layers are:
- DAL Data Access Layer   →   for accessing tables
- BLL Business Logic Layer   →   logic and functions
- SL   Service Layer   →   calling up logic

The DAL encapsulates all accesses to database objects such as tables and views. Its methods regulate the writing, reading and/or manipulation of data. Depending on the complexity of the Forms, the direct table accesses can be transferred to the DAL in a transition phase. However, this step usually proves to be difficult in the case of older Forms applications, as attention has not been paid to the corresponding requirements during the development, e.g. the availability of primary keys.

The BLL records the logic which is necessary for processing an object/domain e.g. customers, orders etc. It can consist of several logically separated units (database packages), so that basic functions can be created, these becoming combined functions in a next step.

The SL is the service layer which accesses the BLL functions, making these available to the outside world. This layer builds the services from components of the BLL layer according to the requirements, and encapsulates these. The SL can take over the tasks of error handling, logging (user monitoring) as well as security (access rights), insofar as possible. At the same time, the user and security concept of the company's SOA strategy should be adopted and incorporated in the SL in the form of patterns.

The DAL and BLL are created by PITSS.CON in the form of templates, predefined program structures and revised by developers for implementation conforming to the strategy.

4.   Procedure

The procedure involved with such a complex project, such as extracting and transferring the business logic to a modern architecture, is outlined in highly simplified form below.

The PITSS.CON tools "Application Analysis" AA and "Application Engineering" AE are used more intensely during extraction and processing of the business logic for the BLL and DAL layers.

By loading all application-relevant data (forms, libraries, database …) into the repository, PITSS.CON is able to introduce a necessary transparency into the source code of the individual programs or even the entire application.

4.1.   Analysis of the application

A parser disassembles the source code corresponding to the SQL, PL/SQL or Java syntax into its components. Based on this information, dependencies and uses of individual components can be determined and depicted. The determination extends well beyond the simple or first use of objects and represents the entire object structure involved in the form of a tracking mechanism.

Applications, functions and their dependencies are therefore visible, and can be documented and processed.

4.2.   Documentation

Applications which have grown over the years usually become rigid and inflexible in terms of their complexity, associated with missing documentation. They can only be modified with a huge amount of effort.

PITSS.CON provides a wide range of documentation or also views of the application, thereby supporting an optimal transfer of source code to the required database structures.

The documentation capabilities in printed, file or data form allow objects to be visualised in their dependencies at any time, and hence understand and transfer them.

Specifications and detailed effort estimations form a background to discussion in the teams, render the workflows more efficient and therefore serve as a basis for cost / time reduction with optimal implementation.

Especially time-intensive processes, such as conversion of a complex application, are provided with the necessary support.

This therefore ensures a comprehensive documentation in/as:
- Project preparation (analysis and understanding)
- Effort estimation
- Project accompaniment (specifications, considerations)
- Quality assurance (before, during and after the project)
- Revision documentation

4.3.   Code reduction / redundancies

Before beginning the development stage, as when converting the application from the existing source code, crucial importance is attached to reducing the application scope.

PITSS.CON provides tool support for the code reduction through:
- Dead code analysis
- Determining code redundancies

4.3.1   Dead code analysis

The dead code analysis identifies source code as well as objects (e.g. tables), which are not used in the application and which can therefore be removed without problem.

Program units are checked to ascertain whether they are used in the entire application. If not, the non-usage e.g. of SQL libraries, PLL's on the level of functions, packages, … and/or on a library level are depicted.
Significant reductions in the application code at up to 30 percent are therefore possible and hence worth pursuing before more complex development work. The time effort is reduced correspondingly.

The reengineering process for applications, the structure and their conversion is optimally prepared.

4.3.2 Determining code redundancies

The further usability of functions and processes is not only a leading concern in software development, but also one of the key arguments for service-oriented development. As a result, PITSS.CON places special emphasis on determining reusable program units.

The analysis methods identify similar or identical program parts. At the same time, similar or identical statements, functions, declarations and assignments can be incorporated as decision criteria.

Core processes of the application are recognised, the similar functions or processes adapted to one another, moved to the database for reuse and written back as a function call-up.

Considerable importance is also ascribed to this functionality after complete transfer of the source code to the database, as such a consideration that takes into account all database objects provides order to the source code again.

4.4. Analysis of the functionality and its complexity

The analysis of the functionality is the next important step when evaluating the Forms-intrinsic logic for a transfer to the database.

The representative analysis results enable a clear estimation for the:
- Nature of the source codes,
- Complexity of the source codes,
- Evaluation of the transfer effort (traffic light, e.g.: green=simple)
- Frequency of use in the application

At the same time, the analysis process can be controlled via parameters in such way that the logic for an object, process or functionality is determined specifically:
- Limitation to: Objects (e.g.: tables)
                                           Modules (forms/libraries/..)
- Limitations to functional logic: Statements
                                           Functional call-ups
                                           Database accesses
                                           Forms build-ins

5. Estimation form the tool use

Transferring a complex application to a new technology is associated with many risks which mostly result in time and cost overshoots or even end in a halt to the project.

The future technology poses some challenges for companies and their employees when it comes to learning, mastering and using it. Consultants and architects who are able to devise such an architecture are required for its success.

The next hurdle not to be underestimated involves implementation of the architecture aimed for. It is decided here, whether you have to begin with the implementation from the scratch owing to the absence of documentation or whether you can derive or even transfer business logic from the enormous investment made in the old application.

If a purely manual implementation is assumed for this project, the requisite knowledge concerning the nature of the logic, its characteristics and dependencies is usually absent.

Tool-assisted automation applications, as offered by PITSS.CON, show the developer the structure and content of his source code, together with all its dependencies. This enables him or her to compile a comprehensive and representative documentation, while deriving specifications from this which lead to a successful and efficient implementation in coordination with the system architects. Even the implementation is supported by PITSS.CON and secured on a lasting basis by accompanying analyses.

6.   Experience

Thanks to its versatile and powerful features, PITSS.CON is the first choice for numerous customers, as a tool that is able to analyse, understand and process Forms applications.

As a result, the latest decisions opting for PITSS.CON are usually made with the aim of maintaining an application more efficiently, developing it further or transferring it to new architectures via knowledge of the logical and functional relations.

Some customer decisions:
- Tieto Enator UK   →   Opting on account of migration and analysis
- AXA PT   →   Evaluation for SOA
- Maersk DK   →   Business logic for transfer to SAP
- Drive Solution UK →   Analysis of the business logic