

ATTRIS Corporation



**PEXA Suite  
Technical White Paper**

Advanced Analysis Methodology to Deploy Enterprise System without Coding

The First Version-01

August 26, 2010

Masanori Yasumitsu  
CEO ATTRIS Corporation

# PEXA Suite Technical White Paper

## Table of Contents

Introduction.....	2
Solution is not Solution .....	3
Change the Process of Application Development.....	3
Seamless Development (PEXA & UML).....	5
None Coding Development Process .....	5
PEXA Architecture.....	7
▪ PEXA Methodology	
▪ PEXA Tools	
▪ PEXA Engine	
Impact of PEXA Development Suite.....	11
Future of Application Development.....	12
About Author and Attrris.....	13

### Introduction

Today's complex and rapidly changing business environment makes it extremely important and yet extremely difficult for management to obtain up to date business information and identify problems throughout the entire company. Daily business processes continue, but timely information about purchases, manufacturing, inventories, shipping, invoicing, collections, etc. in order to optimize results are difficult to obtain if at all. Enterprise Systems, which integrate all group processes and planning in real time to support management decision making, is both the never-ending dream as well as nightmare of Enterprise System Development.

## Solution is not Solution

Historically, the innovation of system development comes from 2 directions, which are new tool development or new method discovery. Tools are mostly coming from programming side for making easier to coding and debugging, and Method are coming more design side to analysis and deployment.

Typical examples of CASE tools were achieved speed of programming and debugging, but replaced by Package software at most of parts and become modification tool beside package. Methodology brings more logical and scientific approach for analysis and design, but somehow difficulty to be along with usual business practice, happen at day by day operations.

These solutions are rather accelerated problems  $N!$  (Factorial) at large complex Enterprise Systems development, because they are not solving fundamental problems.

It is very simple to solve this problem !

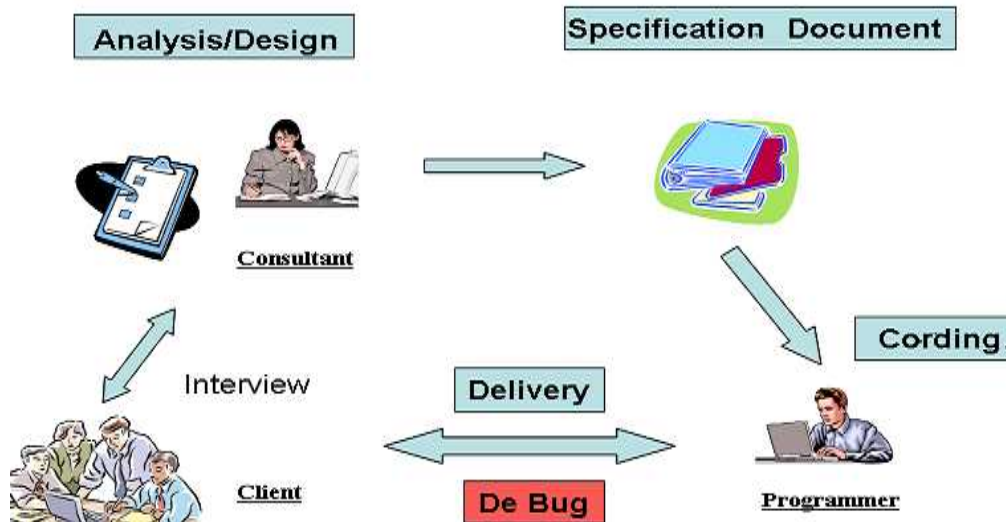
To analyze business and down to execution must be performed consequent methodology and must reflect result of analysis directly to deployment. Otherwise miss matching between design and deployment always happen and twisted like as Chinese Whispers. This simple method never applies any solutions, CASE tool, Package nor UML to Code.

## Changing the Process of Application development

Existing procedures for application development are:

- 1) Analysis about requirements and procedures (As Is); define specifications and requirements (To Be); and then commit the results to written design documents.
- 2) Program the code to the referenced design document and debug the code.
- 3) Test and revise back and forth until completed, given the requirements and specifications in the design document.
- 4) Deliver Finished Application to Client for review and acceptance
- 5) Often back to 3) because the design document does not reflect Client's needs correctly nor program does not code as well as designer expected.

## Existing Development Process



Problems of existing process are:

- 1) No scientific method to create the analysis and design for Enterprise applications. Therefore, no logical documents are easily generated;
- 2) Coding is a highly programmer dependent process and each programmer generates a different style of code;
- 3) No standard procedures exist to easily incorporate changes to the design document and/or obtain feedback from the business user.
- 4) No seamless development process cause miss matching from initial design to final coded application.

These problems cause serious impact not only at the development stage, but also for future maintenance of the Enterprise Application.

Seriousness is increasing as exponentially following complexity of systems, and it is extremely difficult to design and deploy today's Enterprise system.

## Seamless development (PEXA and UML)

PEXA is might be simple, useful, convincing, easy to learn. And there is indeed such a notation for object-oriented Analysis and Design with seamless development environment up to deployment, whose entire set of graphical symbols fits on one page yet covers all the basic object-oriented system description techniques, like widely known UML and which is not particularly applicable at scaling up for the description of large Enterprise systems:

PEXA Methodology is started interview with clients to define requirement, create scenario domain table, and sequence diagrams, as well as UML does.

However UML, different semantics for a class depending on whether its name appears in roman or italics; but at least a programming language, even the worst of languages, is executable! Here you have to learn all this monstrous complexity just to build diagrams of a possible future system.

This brings the question of seamless development. Once we have beautiful diagrams, we will want to build a system, unless the budget has already run out on CASE tools

## None Coding Development Process

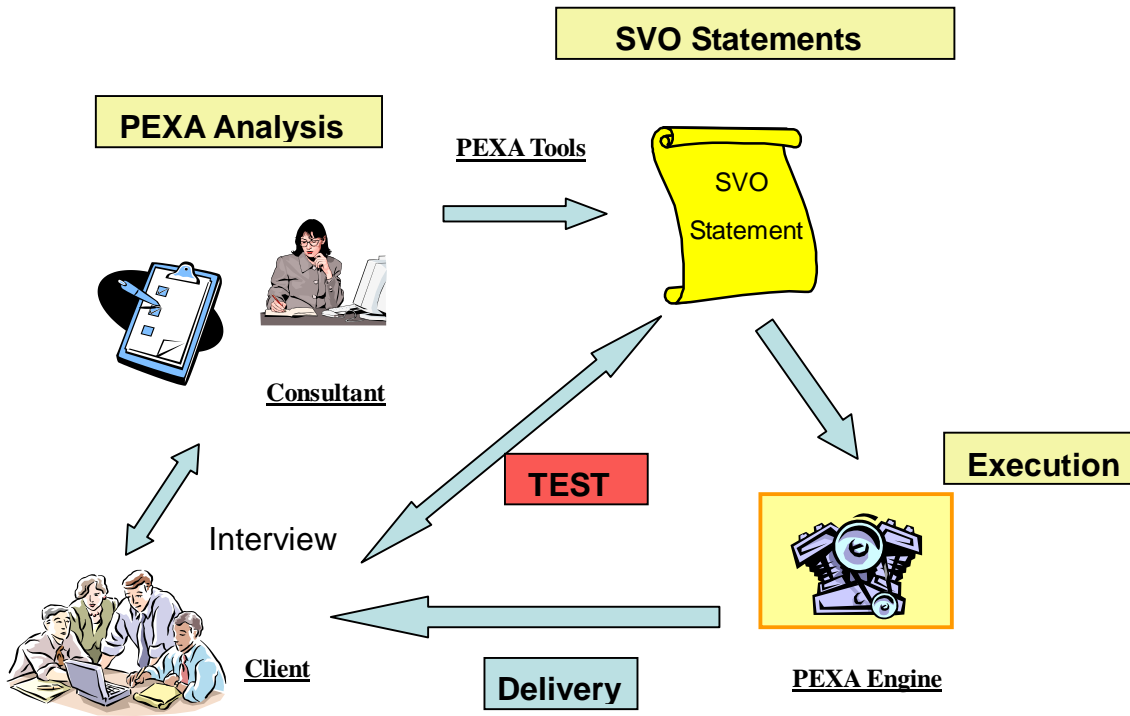
PEXA Suite is designed consequent process from Analysis to Deployment that is the result of analysis and become direct execution of application.

SVO-Statement which generate as result of Analysis by PEXA Methodology are directly executed by PEXA Engine, feature is a so called "Virtual Application Engine".

PEXA Suite does not require programmers to write code or any form of programming. The results of the Analysis/Design stage, written Statement in natural language, are understandable and executable statements describing the procedures desired.

**(PEXA stand for Pluggable and EXtensible Architecture and set of methodology, Tool and Engine as Suite. PEXA is Patent pending and protected by copy right)**

## PEXA Development Process



This absence of programming architecture is a real breakthrough since computers require programming and the generation of code in order to execute. From Analysis to Execution, the application is performed by PEXA's methodological process and eliminates all reliance on individual programmers during application development. This shortens the development time, creates a clean application with ease of maintenance, and allows more focus on the Analysis and Design stage by business managers.

Application must be robust and ensure reusability and extendibility.

But the heavy apparatus of UML all but guarantees that developers who have been able to produce any system description at all will be scared to death of changing anything. For reusable; nothing in UML addresses the challenges of reuse, such as standardization of interface conventions, option-operand separation, and command-query separation and so on. Addition to efficiency UML still requires something to generate code.

PEXA is the first application development platform to achieve all of above

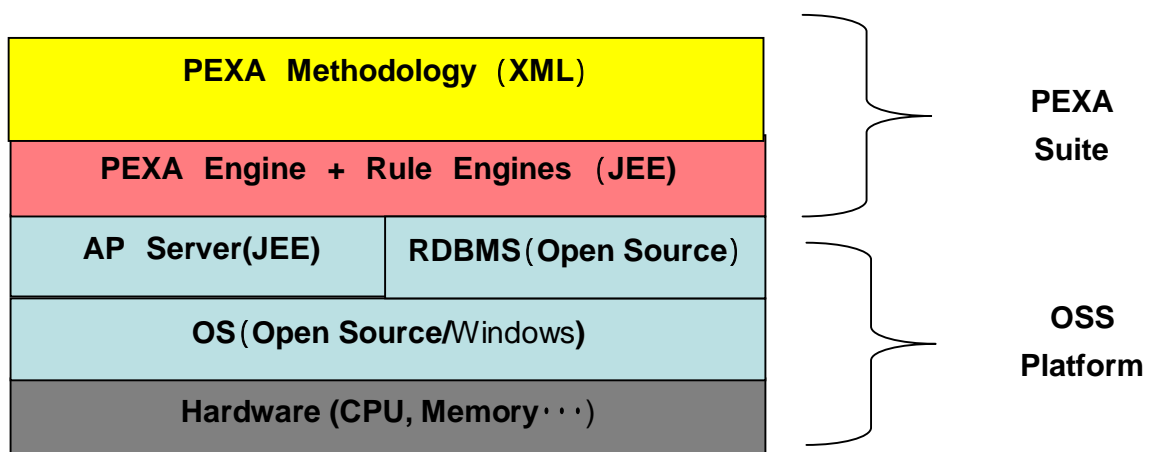
## PEXA Architecture

The PEXA suite consists of three major parts from Analysis to Execution, and covers the entire application development cycle. PEXA requires no programming since it interprets SVO-statements (Subject, Verb, Object) written under PEXA Methodology.

PEXA Tools provide cross-references and verify that the SVO-statements are ready for the PEXA Engine to interpret and execute.

- **PEXA Methodology**  
An analysis and design solution based on RIT Data Profiler
- **PEXA Tools**  
System development tools that create and manage artifacts as well as support project management
- **PEXA Engine**  
An application execution environment based on the artifacts derived by PEXA Methodology

### PEXA Suite Structure



## 1. PEXA Methodology

PRXA Methodology is based on RIT Data Profiler theory.

**RIT stands for Role, Initiative and Target:** **Role** means a team or staff in charge of a certain job in a project; **Initiative** means a set of commands to execute the jobs; **Target** refers to a single, subdivided job, occasionally expressed as an “object” or initiative targets. By using this methodology, PEXA processes data based on methodology of handling individual jobs rather than their contents. This brings the following key benefits for Analysis and Design Applications:

- Benefits
  - Applicable to any type of business
  - Usable by non document writers
  - Verifiable by outsiders (non-experts)
  - Verifiable by dedicated PEXA Tools

### Process of Analysis

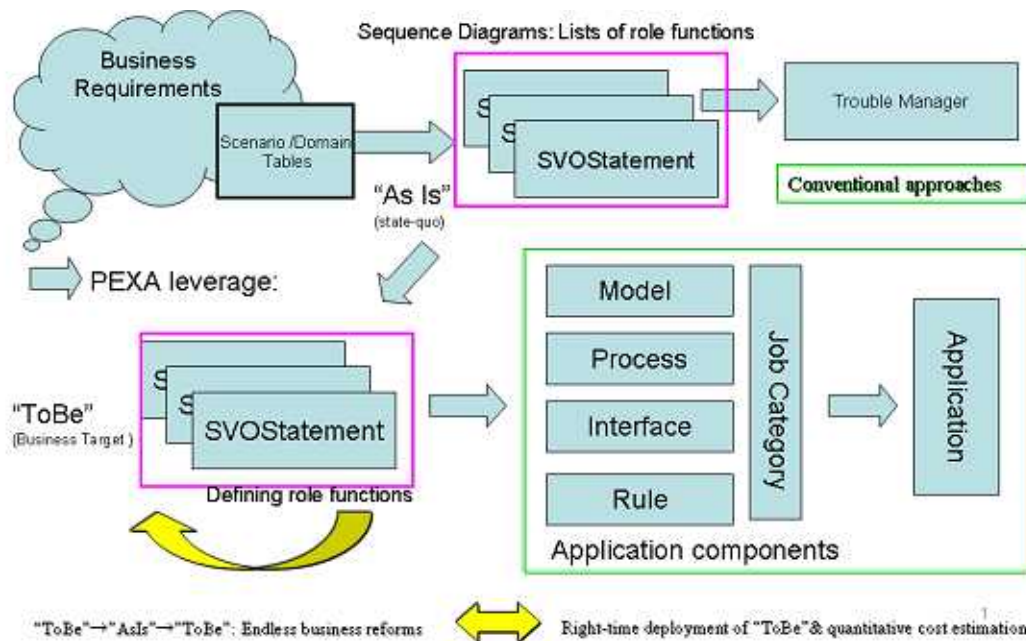
PEXA Methodology focuses on Enterprise System Development and brings consistency process of analysis and design which instruct business flow to computer application.

To analyze current business requirement (AsIs) follows RIT Data Profiler (PEXA's notation rule), to sort out business rules, processes and data handling, and classification them to catalogs. With PEXA tools, Sequence Diagram and Scenario Domain table are generating from those information. Then automatically generate SVO-Statements that define the roles and function in English written from.

Taking SVO-Statements with Trouble list, reviews processes and rules for corrections and adjustment (ToBe). To run prototype is to make sure Application is working correctly and interface is comfortable to users.

Activity Sheet, Model sheet, Task sheet, Rule sheet are also generated as final design documents beside SVO-Statements. Now ready to PEXA Engine for execute Application. Along with PEXA, the Completion of Analysis and Design means that Application development cycle has been completed as well as seamless development, Design to Deploy.

## RIT Data Profiler



## 2. PEXA Tools

PEXA Tools generate SVO-Statements and related documents, and ensure that all parts at the Analysis creation stage are in order before Execution by the PEXA Engine.

PEXA Tools have following key features:

- **Scenario/Domain Table**
  - Categorizes all business requirements either into scenarios (external demands) and domains (resulting in-house jobs)
- **Sequence Diagram**
  - Illustrates how individually allocated "objects" and information (initiative targets) are processed among role members per scenarios and domains
  - Identifies any issues or conflicts faced by each role
- **SVO-Statement**
  - Defines role functions as "Subject(=role)", "Verb(=initiative)", and "Object(=target)"
  - Creates the SVO Statement automatically from sequence diagrams via PEXA Tools

- **Problem Manager**
  - Creates a list of problems or conflicts identified while composing sequence diagrams
  - Allows users to address corresponding and closely related problems or conflicts.

All program processes describe the results of Analysis (As Is), and then review these for Design (To Be) adjustments that may be needed. SVO-Statements are the Output instructions of Job processes (and behavior) created from natural language expressions. These SVO-Statements are the final documentation and output required for Application development. The PEXA Engine takes the statements and executes Applications without the need to program a line of code.

### 3. PEXA Engine

The PEXA Engine is the core of the PEXA 5<sup>th</sup> generation development suite. The PEXA Engine understands SVO-Statements and executes directly. The PEXA Architecture is designed to connect natural language to machine readable code. The Engine is written fully in Java and the connection between a company's operating system platform and SVO- statements works as a virtual application machine.

- **System Requirements**
  - JEE environment
  - JEEAny Application Server and RDB under JEE environment
  - OR mapping feature
  - Stored procedure, XML, CSV mapping
  - Rich client support (Swing)
  - Web service support (JSP,Servlet)
  - Browser support (Java Web Start)
- **Current independent components designs (Bean, Liberially)**
  - Client
  - Model
  - Service
  - Event
  - Print frame
  - Plug in support

## Impact of PEXA Development Suite

PEXA Suite brings a new generation of application development for Enterprises Systems and present 2 industry 1<sup>st</sup> innovations, Analysis Method and None Programming Seamless Development Architecture.

**Analysis Method** is so call PEXA Methodology which will replace the most popular UML system modeling method. UML brings the first modeling method to visualize, to organize and to documentise Applications, and make available for unification of “AsIs” Analysis and “ToBe” implantations. However today’s complex Enterprise System may not applicable for UML , because unless seamless architecture, offset of output of UML to Coding are creating to cause miss matching between design and implementation.

So that Programming inconsistently generate dialogues unless UML directly generate code to execute.

**None Programming Seamless Development Architecture** is eliminating not only for inconsistency problem of UML, but also away from the most labor intensive part of development, Coding and Programming. PEXA Engine read directly Statement generated by PEXA Methodology and deploys to execute its instructions. Other hand UML does not offer consistency process from Analysis to Deployment.

Another huge impact of PEXA is elimination of Coding, which over 60% of time and cost spent for Coding and Programming plus sometimes creating problems by human related. With existing Enterprise applications, managing and maintaining billions of lines of source code is unrealistic, leading over time to neglect and out of date lost code. With PEXA, ease of maintenance and flexibility to evolve are equally important as the speed and lower cost of development. With the PEXA advantages of reduced cost of development, shorter time to development, reduced complexity of modifications and trouble-free maintenance, the cost reduction is estimated to be 25%~90% compared with current coding based Enterprise Application development.

Any project manager knows the challenge in narrowing the gap between the business managers’ knowledge of what analytics are desired and the programmer’s knowledge of what is possible for a given time and cost. This is not to mention the design challenge of anticipating future needs and changes. PEXA is a dream come true. It allows the business managers to go directly to Job Requirements (SVO-Statements) which they know best and see results quickly and dependably.

- Benefits of PEXA Architecture:
  - More time and effort to Analysis and Design
  - Eliminate the time and cost of Coding and programming
  - Eliminate reliance on programmers
  - Achieve consistently high quality application development
  - Benefit from the ease of reading Applications rather than code
  - Bring the application development closer to the business user.

### Future of Application Development

With continuous innovation of hardware allows the creation of more complex applications executed at higher and higher speed. At the same time as business complexity and user requirement are increasing substantially, but software development techniques struggle to keep up. This causes serious problems for complex Enterprise system development and exposes the real limitations of today's method of development. The right way to overcome limitation for application development will be the change weight of time from Analysis to Execution. To achieve this more time should spend to Analysis rather than struggle at Coding stage. Engineers must spend their effort to Analysis stage which only people can, not machines.

Para dim sift is brought by PEXA today.

PEXA allow people, but not always engineers, to spend Analysis and Design and rest of time consuming process are governed by machine.

Methodology never always requested computer engineers, but more likely business person who knows own business process, be able to Analyze own job tasks and generate statement to execute without any coding.

The real end user computing can come true if the Application Development can be done by instruction rather than programming. The non-programming natures of PEXA for Enterprise Application development make available user to build and maintain own application by themselves..

It is the future and benefit bringing by PEXA .

## About Author and Atrris

Masanori Yasumitsu is founder and CEO of Atrris KK. He founded Atrris after compiling a 30+ year track record in computer system engineering at leading high technology companies. He joined several leading system vendors entering the Japanese market: Digital Equipment Corporation, Apollo Computer and Sun Microsystems, serving in each as Sr. Director or Researcher in computer system engineering. He was well known for creating and implementing Java based systems with unique architecture and methodologies. This has led to substantial recognition in the market as one of the leading engineers in the Java world. His career started at a chemical plant as a system engineer for feedback control, and later moved over to Research and development Director at Sun Java Center.

He holds a bachelor of Physics degree from Nagoya University.

Atrris was founded in 2004 as a result of Yasumitsu's realization from 15 years of experience developing Enterprise Systems that there are few systems that satisfy the customer's requirements and even fewer reliable systems. The belief was that this problem results from the gap between business processes and the programming complexity required of hard coding to create the application. Atrris developed the PEXA platform which combines the most advanced analytical methods and frameworks with an execution Engine. PEXA allows evolutionary change of the Enterprise system development, design, coding and maintenance to allow for the realities of today's business needs