

Milan Ignjatovic

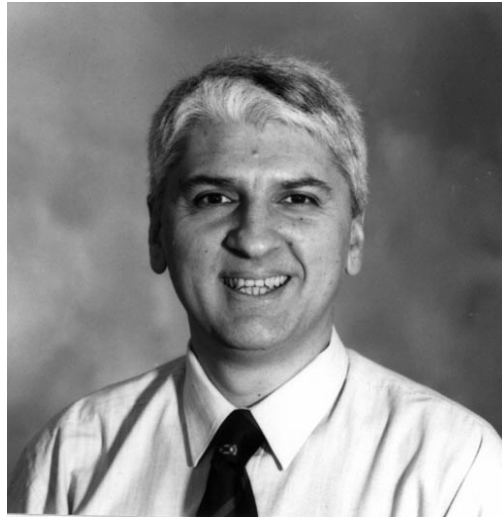
ProSoftwarica GmbH
Technoparkstrasse 1
CH-8005 Zürich
+41 (0) 79 376 59 11
info@prosoftwarica.com

MSc Electrical Engineering
University Belgrade

Postgraduate Studies
Computer Science
University Belgrade

Software Engineering Consultant
Software Engineering Trainer & Coach

E-Mail: mig@prosoftwarica.com



Professional background

2006
Founder ProSoftwarica

1998 – 2006
Zuehlke Engineering AG
Software Engineering Consultant,
Architect, Coach and Trainer

1995 – 1998
Metromec Software AG
Software Engineer,
Project Leader

1989 – 1995
Sulzer Chemtech AG
Software Engineer, EDP Analyst

1986 – 1989
Aerospace Institute Belgrade
Software Engineer / Analyst

1984 – 1986
Research and Development Institute
Belgrade
System Engineer, Acquisition and Control

1984
University of Belgrade
MSc Electrical Engineering

Major areas of expertise

- ✓ Management of Complexity, Abstraction, Separation of Concerns, Decomposition, Model Driven Development, Software Product Lines
- ✓ Requirements Engineering, Use Case & Feature Modelling,
- ✓ Architectural definition and evaluation
- ✓ Software Development Processes
- ✓ Design of distributed Enterprise, Embedded and Real-Time systems
- ✓ Design Patterns, Refactoring Technique
- ✓ Object-oriented Software Metrics
- ✓ Middleware: Corba, Connexis, Tao, Ace, Zen, JacORB, IDL
- ✓ Natural Language Processing: GATE
- ✓ Tools: Topcased, Papyrus, Eclipse, oAW, MetaEdit+, IBM Rational, Rhapsody, Enterprise Architect, MagicDraw, Poseidon, Together, Visual Studio
- ✓ Platforms: VMS, Unix, Windows, Linux, Jbed, VxWorks, PERC, VenturCom, embedded NT, DCS-RTOS(Yokogawa)
- ✓ Languages: UML2, xUML, QVT, ATL, XSLT, Mofscript, AOP, Xpand, Xtend, Check, Java, Javascript, C/C++, Ada, Delphi, APT Graphcet, Modula2, Pascal, Basic, DCL, TPU, Fortran, Assembler
- ✓ Web technology & Portal architecture: Glassfish, WebSphere, WebLogic
- ✓ Application integration & web services
- ✓ Database management systems: Hibernate, OO2RDB mapping, Oracle, Informix, SQL Server, MySQL, HSQL

Project examples

Development of JEE Application Platform

A large Swiss bank develops and operates a massive JEE based platform including a reusable software stack. The objectives are to provide a reliable platform based on JEE technology and achieve a high level of reuse for front-end applications, currently numbering in hundreds. Reuse is understood as a cross cutting concern affecting the full software development lifecycle of in-house components.

Governance of in-house, as well as 3rd party components is also a related concern.

The first task is to assess the current state of the platform with respect to its objectives and propose solutions for overcoming identified problems. This is accomplished by an introductory study of abundant documentation, acquisition of anecdotal evidence and stakeholder interviews. The analysis reveals several emerging, wicked problems, hard to foresee and manage early enough, in order to mitigate all involved risks. The applied governance models based on independent component life cycles without clear ownership and accountability, leads to quality problems. Solution is found in introducing quality gates to synchronise independent lifecycles and guarantee required component quality attributes, thus enforcing service level agreements between applications and their customers. Further problems and solutions include topics such as platform layering, mandatory and optional usage of components, component specification and qualification, conformance to licensing models, deprecation policy, immature interface evolution and obsolete user documentation.

The second task is to provide an architectural solution to extend the front-end application dominated JEE platform, into providing enterprise strength SOA services. Gap

identification is driven by an analysis of missing pieces within the architecture, processes, methods, tools and technologies, currently available on the platform, with respect to managed services and needs thereof. More than 20 stakeholders are interviewed, their input critically analysed, redundancies eliminated and structured into high level requirements and use cases. Documented gaps include OLAs for testing environments, location transparency, dynamic load and capacity management, SLA control (sustainability of service quality attributes), complete service dependency information, service monitoring, service retirement, cost accounting, separation of testing environments, monopolising test environment resources and end-to-end business transaction monitoring. Architectural solution is accompanied with a rationale for refactoring, defining new centralised processing components aiming to factor out redundant or similar processing across the entire enterprise.

The third task is the elaboration of a concept and architecture defining a standard solution for distributed caching on the JEE platform based on Oracle Coherence. The need is driven by a number of new applications migrating to the JEE platform that require distributed caching support. Initially, a number of common use cases are identified shared by the migrating applications. The elaborated architecture extends the existing platform architecture to satisfy the identified use cases. The result is a fast and smooth migration of a number of applications to the JEE platform with standard platform support for distributed caching functionality.

Technologies: JEE, Oracle, WebLogic, Coherence, Ordering Tool, Eclipse, Maven, UML/MOF, Windows, monitoring, logging and incident management.

Emerging System Paradigm

The effort for human management of N evolving components, using conventional technologies, is non-linear. The conceptual idea behind the Emerging Systems Paradigm is to linearise such task complexity.

ESP splits a single complex executable system into a set of "super-loosely" interacting, individually manageable and executable components. ESP captures the evolving system interaction, resulting from natural system evolution, into a separate interaction model, enabling the "super-loosiness" feature of the system.

ESP processes the "super-loose" component set programmatically to deliver the executable target system. Thereby, the target system has an emerging character to the individual executable system modeller. ESP is equipped with following features:

- Non-invasive modification of source models by multiple extension models.

- Executable target model is composed, both statically and dynamically from several input executable models and interaction models.

- The composition process is driven by the interaction model and business rules.

- ESP scales well by enabling programmatic processing

- ESP is robust, fully automated, therefore well suited for human exploitation.

Therefore, ESP enables simultaneous management of N interacting components to be reduced to the management of each component in isolation from the others, thus linearising task complexity for the system developer.

Technologies: J2EE, Eclipse, Topcased, MOF, UML, UML Profiles, XML, Mofscript, Business Rules, Swing, TouchGraph, SQL, MySQL, Windows.

Web Service and Confluence Solutions

A large Swiss bank has demanding needs for Web Service definition and realisation. This need is driven by the high complexity of the underlying communication infrastructure and heterogeneous subsystems, varying in technology from mainstream to legacy. The existing software solution is not well suited for this task because it does not automate the complete workflow. For example, during web service specification and artefact creation, the engineer is forced to execute repetitive, tedious and error-prone manual tasks resulting in schedule delays and realisation bottlenecks.

In order to alleviate these problems, an automation solution is realised based on custom XSD and WSDL artefact generation. A central UML repository is realised by porting and re-engineering the existing models into well-structured service packages. Each service package and elements thereof are furnished with meta-data which is defined within UML profiles. Artefacts, XDS and WSDL service descriptions, are generated using type-based recursive drill-down algorithms by either referencing or de-referencing elements according to custom requirements of the target subsystems e.g. de-referenced XSD element description for Cobol Copybook generation.

For presentation purposes, models are programmatically harvested and only meaningful i.e. meta-data defined end-user information is plucked. Further down the processing chain, this information is packed into SQL statements, typically Oracle MERGE statements, which are executed against an Oracle Database to populate the Confluence DB schema. In this way, the modelled web services are shared on the intranet with the end-users using a Confluence system. The process of information sharing is automatic and results in the generation and execution of many thousands

SQL statements. End-users access the published Confluence information to check and verify existing web service specifications or to guide themselves during new web service definition.

Technologies: proprietary software development process, Enterprise Architect, Eclipse, Topcased/Papyrus, XSLT, UML and UML Profiles, ATL, Mofscript, SQL, Oracle, Wiki.

Enterprise Portal Solution

A large Swiss bank engages in a project aiming to deliver a web portal system as a solution for its numerous external asset managers.

The task is to define and specify the enterprise system architecture by elaborating the prescribed documents and getting them through the internal software development process.

Furthermore, a kick-start with the implementation team is to be accomplished and the project safely guided into the realisation phase.

Technical specifications such as non-functional requirements, solution blueprint and design, infrastructure design, security and test concepts are elaborated. Additionally, disaster recovery, availability, scalability and monitoring concepts are also elaborated to the required detail.

Following subsystems comprise the architectural solution: secure entry servers, load balancers, WebLogic cluster, WebLogic portal, portlets and business services subsystems. Legacy subsystems include the Host backend, output management subsystem as well as archiving and monitoring subsystems. To complete the scenery, Identity and Access Management subsystem is extended to handle extravagant external company requirements such as definition and management of team hierarchies with client-to-team assignments and 2 user-to-team relationships. LDAP and Active Directory subsystems are reused.

Special attention is given to security aspects such as single sign-on, DMZ, protocol isolation, firewalls, denial of service, penetration tests and vulnerability detection based on automatic scanning and testing tools.

Technologies: proprietary software development process, UML, WebLogic Application Server and cluster, ALUI Portal, MSSharepoint, Siemens DirX, IBM Rational AppScan, IBM OnDemand, Oracle, Eclipse, Java, XML.

Client Reporting Solution

A large Swiss bank requires a reporting solution to be built to complement its front end suite of applications.

The development life cycle starts with requirements engineering, technical specifications, architectural definition and solution design. Large reporting volumes e.g. at the end of the month processing, are recognised early enough in the project and formally recorded in the non-functional requirements document. Elaboration of a multi-threaded working prototype follows and proof of concept for the proposed architectural solution is obtained with guaranteed throughput. Software construction is carried out with a team of 3-5 developers and 2 testers over a period of 6-9 months.

Technologies: proprietary software development process, asset/portfolio management system Odyssey TripleA, output management system, archiving system, monitoring system, UML, WebLogic, WebSphere, WebSphere Message Queue, IBM OnDemand, Oracle, MSAccess, Eclipse, Java, XML.

Test Automation

A large and renowned insurance company decides to publish its 500+ products on the internet in order to enable its agents to quickly access

and accurately service their clients. In the past, testing of the 500+ products has been done manually and was plagued with massive effort estimates, scarce resources and bottlenecks. It was clear to the test department that going on the web required substantial practice changes. To the company architects and managers faced with this growing problem, test automation intuitively resonated as a viable solution, though no one on the team had direct experience with this field of practice. Before launching into a full-scale change initiative, they decide to expand their understanding of basic automation principles with assessment, training and consulting by an external company specializing in test automation best practices i.e. ProSoftwarica. During assessment, need is recognised for meta data that would drive both the development and test activities. The meta data available was obsolete, so the team decides to consolidate and extend the definition of the meta data with test data. A testing tool is extended to handle the meta and test data and used to drive the tests in a fully automatic mode. Within 5 weeks, positive results motivate the team to go further and provide a completely automated testing solution. The fact is, that after 5 months of automatic testing, defects have been uncovered that existed within the subject-under-test applications, for years. Technologies: RUP, UML, Java, XML, Rational Functional Tester.

Reservation, Visitor Management, Operations and Workplace Automation System

Workplace workflows such as meeting/conference room reservation, visitor management, conference operation control and workplace sharing are fully automated. The

architecture is based on a product line consisting of core services integrating the 4 above mentioned products. Since each product is based on the same core, clients upgrade their systems simply by purchasing additional products. All products are made highly configurable using properties and business rules. Each client may define and maintain their own business rules e.g. multi-tenant capability and deploy them instantly. This capability promotes client independence by reducing the possibility of vendor lock-in. The realised system interacts with existing client/server applications e.g. Outlook/Exchange or Lotus Notes Domino environments without invasive modification of these systems i.e. data synchronisation is done periodically using server APIs. This eases installation, maintenance and training. Furthermore, Building Automation Systems, Access Control Systems, ERPs, Billing, Back-Office and other end user systems are integrated. Additional features are: 3000-5000 users, 300 simultaneous users, horizontal scalability within a server cluster, load balancing, fail-safe, single sign on, software as a service i.e. hosted version, web services and remote installation, test and support. Responsibilities: CTO, Chief Architect, Team lead of 8 software developers. Technologies: RUP, UML2, Windows, Java, Jboss, MySQL, XML, Clustering, Business Rules.

Innovative algorithms for the foreign exchange market

Highly computerized, programmatic and algorithmic trading approaches in the foreign exchange domain have become a dominant market force. The client is an expert in the foreign exchange market and is interested in developing a set of innovative prediction algorithms. Models have been elaborated for the proposed algorithms based on client's

proprietary know-how. Data modelling of persistent entities satisfies often changing and creeping requirements typically found in exploratory and prototyping applications. Complexity has been contained using flexible structures for hosting of extremely large data volumes. Calculation intensive transformation operations are readily applied, extended and modified according to constantly changing requirements.

Technologies: RUP, UML2, Windows, C#, .Net, MetaStock, Metalib.

Multi-server capability for a host-based asset management system

A large multi-user asset management system used in Banking, Fonds, Insurance and Investor markets comprises a number of tightly coupled software applications deployed on a single multi-processor host machine. The vertical scalability of this single host machine has been exhausted to the point where it cannot be scaled any more i.e. by replacing existing multi-processors with more powerful units or by permanently adding more memory resources.

The proposed architectural solution enables the system to scale horizontally using a high performance networked cluster consisting of a dozen of medium sized server machines.

Implemented extensions to the existing system include: efficient load balancing, fault tolerance e.g. reliable fail-over scenarios and real-time cache synchronisation mechanisms. Cluster communication is based on high performance computing technology.

The resulting system is able to accommodate more users and offer an enhanced quality of service.

Technologies: RUP, UML2, Windows, Linux, Solaris, Java and MPI Parallel Computing Technology.

Preparation and evaluation of a tender for a new generation of Self Service Point of Sale - Billettautomaten

A large Swiss railway transport company prepares to develop a new generation of "Billettautomaten" in cooperation with one of 5 competing hardware and software suppliers. The supplier of choice is to be evaluated based on the carefully elicited set of requirements and evaluation criteria. The task is to identify and formulate a tender consisting of the necessary requirements at a specific level of abstraction which are to be implemented by the hardware as well as software supplier of choice. A hardware abstraction layer is identified with a number of interfaces satisfying the necessary requirements. After legal approval, written documents are sent out to the competing suppliers. The received feedback is analysed, questions answered and guidance provided until all details concerning the hardware abstraction layer have been clarified. The final evaluation of returned documentation identified clearly the preferred suppliers for the project.

Technologies: RUP, Requirements Engineering, UML2, Windows and Linux (Debian).

Informix database systems

Analysis, design and implementation of enterprise database systems using 4GL database technology based on the relational DBMS system from Informix running on Novel servers.