

Modernizing Enterprise Systems

Enterprises seek to exploit Digital Technologies for Agility

TATA
CONSULTANCY
SERVICES

RD Naik
Vijayalakshmi Gopal

IN BRIEF

Modernizing enterprise systems sounds glamorous, but it is hard work beset with risk. Enterprises have modernized for varying reasons in the last four decades. The new compulsions are digital: to leverage multiple digital channels by exploiting the new digital technologies, and to make the systems agile.

TCS Research has explored enterprise transformation for three decades and delivered a number of mature tools. Reverse engineering tools provide on the one hand, a macro-level analysis of the software system—sub-systems, call-graphs, database components, UI components, cross-references, and on the other, a micro-level view of the programs—variable cross-references, data slices, program slices, control-flow dependences. Human-assisted methods woven around the reverse engineering tools are in a mature stage of use in TCS through its TransformPlus tool-suite for addressing business problems in the space of modernizing legacy systems and reengineering of applications.

Our future work will focus on business-knowledge-models driven transformation of enterprise source system to the targeted COTS product/package, and incorporating machine-processable specifications of business products, business processes, and business rules more centrally in the forward engineering path.

“Enterprise Modernization” is a glamorous term for an unglamorous job. No CIO wants it on his watch. It is often called for when the CEO is getting some grand fittings in the ballroom, and there is a short circuit, to use an electrician’s metaphor. The CFO will not pay for rewiring

but only for some duct tape and junction boxes. The CIO takes a couple of stooges to the dark and damp meter room with barely a tester and rubber slippers and tries to peer at the problem amidst a tangle of wires. The architect’s electrical blueprint is of no use,

Fact File

TCS Research: Smart Transformation of Software Services

Outcomes: Business Rules Mining framework, Business Rules Modelling and Verification, Match and Gap analysis for Business Rules, Business Rules Mining from Domain-specific Documents, Synthesis Framework for automated Data Transformation Rules

Principal Investigators: Ravindra Naik, Raveendra Kumar and Pavan Chittimalli

Academic Partners: IISc Bangalore, IITB Mumbai, IIT Gauwhati

Techniques used: Modelling of Business System operations, Formal methods, Model Discovery, Data-flow and Control-flow analysis, NLP and Machine Learning

Industries benefited: Any business software system to be replaced with a product or package or creating new custom application(s)

Patents: 4 filed, 1 granted in India and US

Papers: 9 conference papers (Tier 1 and 2)

because there have been too many changes and no one can remember the notation anyway. A patch-up job follows, and is announced as modernization. Sad but true: we do not know enough about software evolution, and even less about organizational evolution to keep our IT systems ever fresh and agile for the business.

The risks before modernization

Whatever may be the objectives, the transformations are trapped with multiple risks. Some of them are as follows:

- The enterprise system has many subsystems, the origins and functions of many are not clear.
- Acquired systems over mergers have many similar processes; some processes seem to be around serving no apparent purpose.
- A good bit of documentation that may explain the

requirement and technology is obsolete, if not missing.

- Vendors that supplied a generation of technology have shut shop.
- The skills with which old systems can be tweaked are not available.
- There is fear of what will break when something is touched.
- The biggest unknown of all: the scope of modernization. You cannot say with certainty, at some level of technical detail, what is the best level of modernization; how long it will take, how much it will cost, and how long it will hold as "modern".

This is why the capability of a reverse engineering tool becomes critical. Reverse engineering tools provide, on one-hand, a macro-level analysis of the software system (for example, its sub-systems, call-graph, database components, UI

.....

Automated annotations based on domain vocabulary for code statements are the key for extracting business rules from source code of legacy implementation.

.....

components, and cross-references), and on the other, a micro-level view of the programs (variable cross-references, data slices, program slices, and control-flow dependence). Human-assisted methods woven around the reverse engineering tools are in a mature stage of use in TCS through its TransformPlus tool-suite, primarily for addressing business problems in the space of modernizing legacy systems and reengineering of applications.

Modernization approaches—today

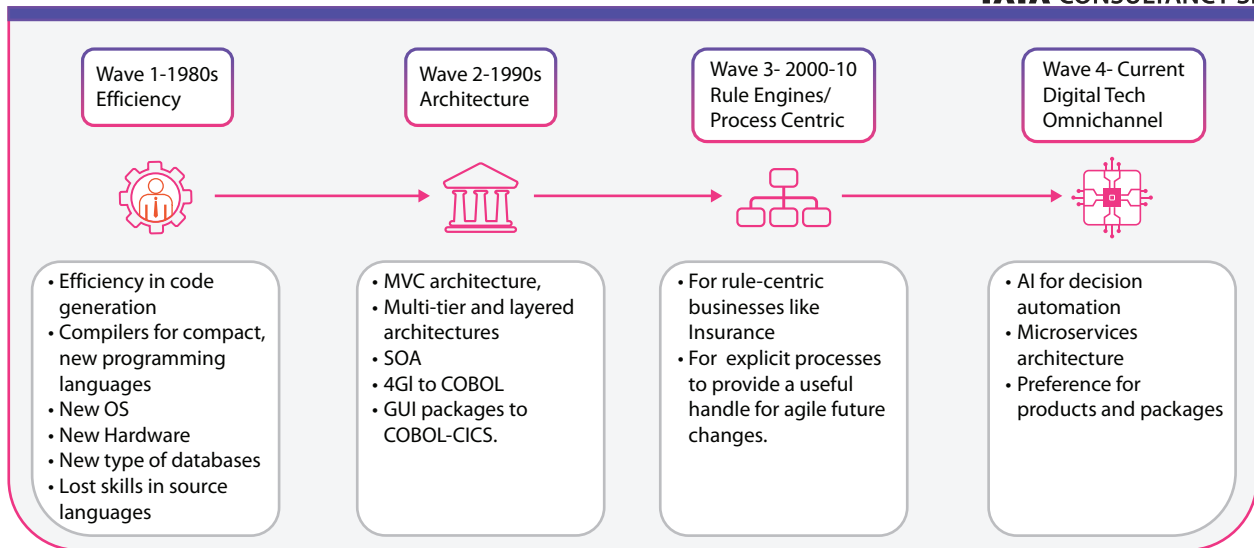
Transformations for enterprise modernization initiatives can be visualized as a combination of:

- Reverse engineering to capture the business knowledge and software design of the existing legacy system: Primarily, the business knowledge consists of business products, business logic, and data expressed using the business vocabulary. The design-level building blocks can be the user interfaces, messages, services, validations, and computation logic along with database and file accesses. An interesting aspect is to visualize each of these as patterns, which can be classified as architectural or design patterns, and search/mine for them in the source code using program analysis techniques.
- Mapping the business knowledge and design so captured to the desired, target architecture: This is a tricky step, involving lot of mental mapping. The captured business knowledge will usually need modifications to embrace efficiency

and simplicity. In fact, the definitions of the design-level building blocks captured in the reverse engineering phase are influenced, to an extent, by the target architecture. Mapping the source building blocks to equivalent blocks in the target is possible using semantic knowledge of the target architecture.

- Forward engineering to create an implementation that reflects the modified processes and logic as per the target architecture: Depending on the choice of target architecture and technology platforms, the forward engineering can involve a combination of code transformation, code generation, and manual code writing. In the case of modernization, the new implementations are created largely by code generation and partly by manual coding. In the case of re-architecting, the new implementations are a combination of code transformation and code generation.

TCS' MasterCraft™ TransformPlus has its origins in programming languages and model-driven development research in TCS' first research lab, Tata Research Development and Design Centre (TRDDC) at Pune. The TransformPlus reverse engineering toolset is based on program analysis, and helps understand systems at the macro level and programs at the micro level. TCS has used this toolset extensively to create documentation and extract models along with specifications from existing systems so that they can be rebuilt without going on the traditional route of specifying requirements. It works for all the



Drivers of Waves of Modernisation

major programming languages in which legacy applications are written, and can be customized for newer technologies.

By bringing in more intelligent and rich user interfaces, the toolset enables users to extract the business rules with a human-in-the-loop process. By matching patterns with predefined templates, the reverse engineering toolset extracts business rules buried in legacy code. Domain-based templates are created by functional experts for different domains and have grown robust by learnings from years of deployment. Leveraging this contextual knowledge, the user can minimize manual effort that a domain expert has to put in, reduce errors, and increase productivity. The business rules so created represent the core business knowledge of the domain as discovered in the legacy application. Especially for enterprises that are primarily rule-driven (for example, insurance), the business rules form the core specifications for migrating the implementation to a third-party rule-engine.

The MasterCraft™ Transform-Plus Reverse Engineering toolset also

depicts the complex landscape of applications in terms of the design-level building blocks in a comprehensible form. The toolset generates very useful analytical information about the software system, some of which consists of:

Inventory statistics: Provides components/lines of code distribution, online/batch Distribution, complexity metrics, hotspots and application interface details

Control flows: Enables the view of complete application control flow starting from scheduler to job to program/class to paragraph/methods or from online transaction to program to paragraph/method level; provides statement-level flowcharts for each paragraph/method

Cross referencing and layouts: Various reports on different types of cross-references are provided. All screen/file/table layouts are generated; enables the complete dependency analysis within different type of application components

Application documentation: Automatic generation of technical

program specification documents; enables the annotation/tagging for business and domain knowledge on top of the technical documentation to make it a program specification

Business rules: Automatic extraction of different rule groups and rule sets based on control flow sequence. Makes use of a flexible framework for configuring different parameters for business rules templates, and auto population of the business rule parameters based on domain value mapping; provides interfaces for technical/business description addition with classification and review options

Impact analysis: Comprehensive and interactive impact analysis workbench for the component level impact/dependency analysis; enables variable-/attribute-level tracing through complete data flow analysis capabilities

These are used in two contexts:

- Semi-automated approach to transform a legacy application to a service-oriented, architecture-based (SOA) three-tier, or micro-services enabled either as on-premises/cloud native application. It exploits a model-driven code generation approach for generating code from a few design-level building blocks that are expressed in the form of models.
- Architecture-level program understanding of the application that facilitates its enhancements and bug-fixing.

TCS MasterCraft™ TransformPlus Forward Engineering Edition accelerates monolithic/micro-services-based development which allows enterprises to handle business changes in an agile way

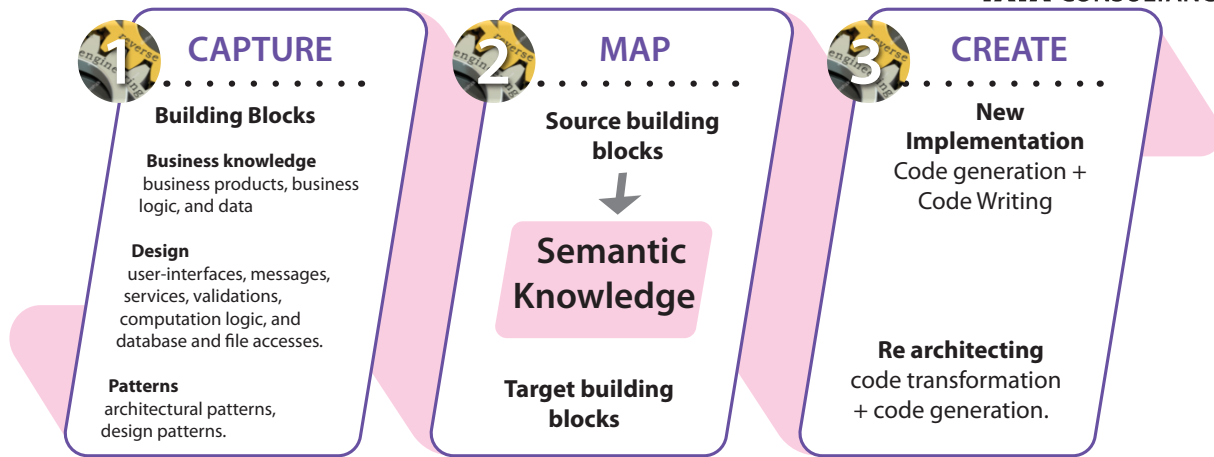
as well as scaling infrastructure automatically to handle increasing volumes. Its features include:

- Template-based responsive UI generation on all major devices and browsers
- Continuous technology upgrades with automated code generation from models supporting the latest technologies, frameworks and libraries
- Application programming interface (API) creation, versioning, and documentation with a Wizard-based approach for API-fication, in addition to documentation and event-based inter-service communication
- Packaging of micro-services and GUI using lightweight Docker containers deployable on all major clouds with auto scaling by help of Kubernetes
- Multi database support for relational database management system (RDBMS) and NoSQL databases using Java persistence API (JPA) for polyglot persistence
- Supports unit testing, service testing (JUnit) and end-to-end testing (consumer-driven contracts)
- Quality of generated code in-line with industry standards

TransformPlus also enables migration of existing applications in an optimized way to new technologies, platforms and architectures in a highly automated way. Some of the migration capabilities include:

- Moving from legacy languages like COBOL to new age languages like Java

.....
 Machine-
 processeable
 Business Knowledge
 is the basis for future
 transformations.



Reverse Engineering

- Moving from monolithic to SOA-based architectures
- Moving from legacy data stores to relational/big data databases

Given the current maturity of the transformation technology, the business rules and design-level building blocks that are extracted and identified in the reverse engineering phase are only loosely coupled with the forward engineering phase. Human interpretation is the key to convert the discovered information into specifications for the forward engineering phase, for modernizing and re-architecting the enterprises. These limitations are one of the drivers for future work. Another key challenge is the formalization of business rules and other missing business knowledge elements. Linking these knowledge elements and the design elements into the forward engineering phase is another important challenge.

A take on the future

We observe that recent transformations are largely aligning with the trends of:

- Exploiting digital channels and providing personalization
- Implementing micro-services and deploying in the cloud

Though the standardization of platforms and technology stacks continue to be important objectives of transformations, they are now complemented by the *agility* that the transformed applications and systems are expected to acquire and illustrate. Towards this objective, the *business knowledge-centric development* and *SOA* can be seen as an alternative and credible path forward. Incorporating machine-processable specifications of business products, business processes, and business rules more centrally in the forward engineering path can provide the all-important traceability between business knowledge elements and software design models, leading to a novel change for the design and development of agile systems.

Another important trend is the inclination of enterprises to replace their custom systems with commercial-off-the-shelf (COTS) products and platforms. The COTS products are configurable using parameters. For such replacements, we believe that the business knowledge models can play an important role. A match and gap analysis between the business knowledge models of the source system and COTS product can help identifying correspondences between source features and product features and can also

Integrating Business Knowledge models in the forward engineering path is critical for design and development of agile systems.

detect missing features in the COTS product.

For meeting the above objectives, we are exploring multiple approaches and techniques:

- Combining formal methods and software engineering for exploring machine-processable and executable representations for business knowledge—business rules, business processes, and business products, as well as exploring methods to *rationalize* them
- Scavenging domain-specific vocabulary (data) entities and potential relationships between them from database schemas and source code of software systems
- Human-assisted mining of business rules from requirements or strategy documents using neuro-linguistic programming
- Automating the extraction of business rules from source code of legacy applications with minimal human assistance, using precise and scalable program analysis techniques
- Human-assisted mining of process models from logs and traces of real systems in execution
- Creating business product models from COTS product configuration descriptions
- Combining NLP and formal methods to compare a pair of machine-processable specifications for identifying the matches and gaps between them
- Using an input-output-examples-driven program synthesis approach for building parts of transformation tools

Sorting the tangled legacy systems of an insurer

A large insurance and financial company from North America has been in the business for nearly 175 years now. It has grown both organically as well as through mergers and acquisitions. An early entrant to computation, its systems had grown with many additions in different phases. The folks who built the core applications on mainframes were long gone. There was no documentation. In a world where nimble start-ups were offering exciting products, personalization, real-time services, omni-channel experiences, the company felt that it was

slow to react and mired in complexity.

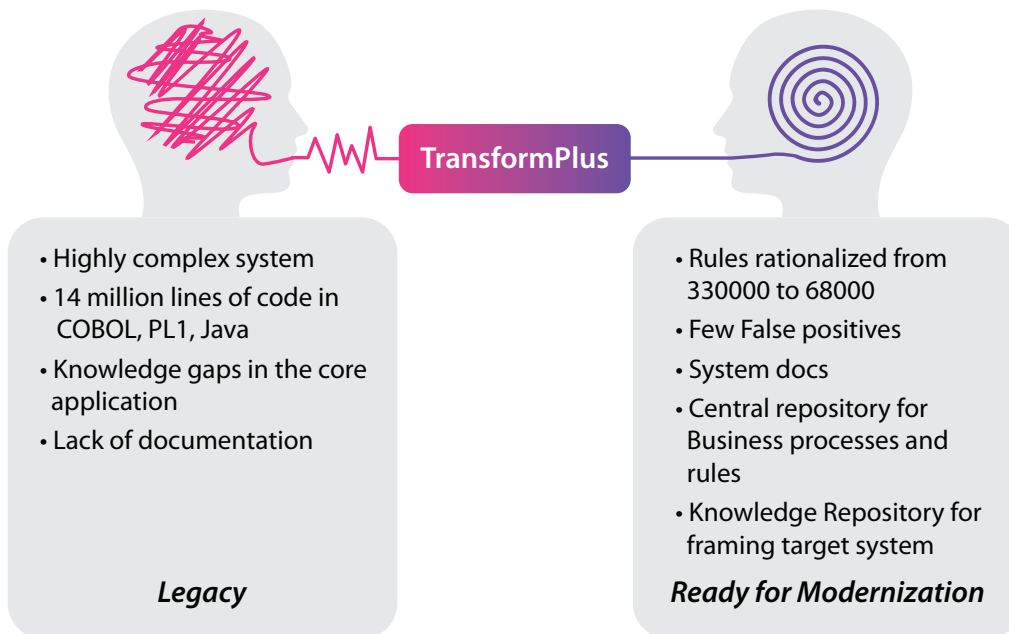
The customer was looking for a large-scale transformation of their entire IT system. The insurance property and casualty (P&C) systems were identified for the first wave of transformation. Its core applications had 14 million lines of code in COBOL, PL1, and Java.

Since there were many gaps in knowledge about the core systems, the customer wished to build a knowledge repository comprising the business rules and technical details of all P&C applications. The repository was also used to help with maintenance during the transition period.

The Application Analyzer, a part of TransformPlus, was customized to better fit the customer's requirements. TCS was able to:

- Identify business rules embedded in the code with a high degree of accuracy
- Rationalize to rules extracted from 330,000 to 68,000
- Generate system documentation
- Maintain a centralized repository for business processes and business rules for impact analysis and reengineering
- Use the knowledge repository to frame the requirements of the target system

The customer gained the ability to go forward to a new technology platform armed with complete information of the existing system.





RD Naik

RD Naik is a Principal Scientist at TCS Research and Innovation. His vision is to evolve a universally acceptable software evolution method, particularly focused to build agile software system for enterprises. With a focus on transformation of software systems as an important business problem, his approaches of exploration include modelling and verification of machine-processable specifications, mining of specifications from structured and un-structured data, documents and source code of software system, techniques for reasoning about specifications, and traceability between specifications and implementation of software system. Some of his areas of interest consist of software abstractions and application architectures, program analysis, formal methods, NLP and ML techniques.



Vijayalakshmi Gopal

Vijayalakshmi Gopal (Viji) currently heads the TCS Tools Group (TTG), which strategizes, develops, deploys, and supports intelligent automation products for application development and delivery. Viji is also the business owner for Jile™, an Agile Dev Ops product from TTG for modern application delivery. TTG takes the products to the market under the brand "TCS MasterCraft™", which encompasses 3 products: DevPlus, TransformPlus, and DataPlus. She has 32 years of experience in IT industry.



All content / information in Modernizing Enterprise Systems is the exclusive property of Tata Consultancy Services Limited (TCS) and/or its licensors. This publication is made available to you for your personal, non-commercial use for reference purposes only; any other use of the work is strictly prohibited. Except as permitted under the Copyright law, this publication or any part or portion thereof may not be copied, modified, adapted, translated, reproduced, republished, uploaded, transmitted, posted, created as derivative work, sold, distributed or communicated in any form or by any means without prior written permission from TCS. Unauthorized use of the content/information appearing here may violate copyright, trademark and other applicable laws, and could result in criminal or civil penalties.

TCS attempts to be as accurate as possible in providing information and insights through this publication, however, TCS and its licensors do not warrant that the content/information of this publication, including any information that can be accessed via QR codes, links, references or otherwise is accurate, adequate, complete, reliable, current, or error-free and expressly disclaim any warranty, express or implied, including but not limited to implied warranties of merchantability or fitness for a particular purpose. In no event shall TCS and/or its licensors be liable for any direct, indirect, punitive, incidental, special, consequential damages or any damages whatsoever including, without limitation, damages for loss of use, data or profits, arising out of or in any way connected with the use of this publication or any information contained herein.

©2019 Tata Consultancy Services Limited. All Rights Reserved.

Tata Consultancy Services (name and logo), TCS (name and logo), and related trade dress used in this publication are the trademarks or registered trademarks of TCS and its affiliates in India and other countries and may not be used without express written consent of TCS. All other trademarks used in this publication are property of their respective owners and are not associated with any of TCS' products or services. Rather than put a trademark or registered trademark symbol after every occurrence of a trademarked name, names are used in an editorial fashion only, and to the benefit of the trademark owner, with no intention of infringement of the trademark.