Migrating Legacy Spreadsheets-based Systems to Web MVC Architecture: an Industrial Case Study

Domenico Amalfitano
Anna Rita Fasolino
Valerio Maggio
Porfirio Tramontana

Giancarlo Di Mare
Ferdinando Ferrara
Stefano Scala

Context

- Hardware-In-the-Loop (HIL) testing of Electronic Control Units (ECUs).
- HIL testing is used as a validation technique.
  - Sequences of test cases, named *Test Objects* are executed in a hardware simulation environment.
  - Test Objects are made by *low level instructions* that define:
    - the signals to be sent as input to the ECU Pins;
    - the corresponding expected values measured at the ECU Pins.

Context

- FIAT Test Engineers exploited an Excel-based tool to quickly develop high-level testing specifications, named *Test Patterns*.
- Test Patterns were automatically translated in Test Objects.
  - By means of a lookup table embedding the mapping between high-level operations and the set of low-level instructions.

Context

- Test Patterns was defined by means of a predefined *Excel template* file organized in seven different sheets.
- Each template embedded:
  - the data model,
  - the business logic functions (by means of Macros implemented in VBA) aiding the testers in the data entry,
    - Functions were conveyed by a set of graphical interface components.

Spreadsheet Based Information System Issues

- Template has been adopted to produce the Excel Test Pattern files constituting the whole HIL Spreadsheets-based Information system.
- Information System made by a set of *30,615 different Excel files*, each *containing on average 2,700 data cells.*

Spreadsheet Based Information System Issues (Replicated Data)

- Information System presented a high rate of replicated data, due to:
  - *scattered data* among multiple spreadsheets,
  - underlying *model lacked* any *information about related data*.
- As an Example:
  - almost the 20% of data cells recurred more than 1,300 times,
  - almost the 50% of them recurred more than 100 times.
- Issues for
  - the maintainability of the information system,
  - the data reuse

Spreadsheet Based Information System Issues

- Business logic functionalities were injected in every Excel file via the template with inevitable code replication.
- Poor quality of the VBA code:
  - no meaningful variable names and comments were supplied to ease its comprehension. More than 8,000 LOC with no internal comments;
  - identifiers of 174 widgets over 183 were left unchanged with their default meaningless names (e.g., *Label1, Button2)*.
- Low maintainability of the business logic information system.

Spreadsheet Based Information System Issues

- No concurrent data access mechanism was provided (i.e., Excel does not support it).
- The low spreadsheet development costs corresponded to expensive delays in the testing process.
  - Due to frequent runtime exceptions of the Test Objects caused by invalid input values introduced during the Test Pattern editing steps.

Proposed Migration Process

- Aimed at developing a new information system that overcomes all the issues.
- The new information system:
  - is a Web application, based on the Model-View-Controller (MVC) architectural pattern,
  - exploits a Relational Database to implement the data persistence.

Proposed Migration Process

- Had to accomplish the following specific goals:
  - no data and functionality provided by the previous system had to be lost;
  - backward compatibilities with the Test Objects generation process had to be guaranteed;
  - integrate a concurrent and distributed access to the data;
  - provide a new rich user interface equipped with assistive technologies aiming at reducing compilation errors and at improving the system usability.

Proposed Migration Process

- Was performed in three main steps.
  - Reverse Engineering of the legacy system
    - *Data Model Reverse Engineering*
    - *User Interface and Business Logic Reverse Engineering*
  - Reengineering
  - Validation

Proposed Migration Process
Data Model Reverse Engineering

- The step was performed to the aim of mining the model of the data stored in the legacy system.
- In the legacy system:
  - All the Excel files composing the legacy Information System inherited their structure from a common template,
  - data concerning the same topic in a single Test Pattern file were grouped together in rows or columns separated by empty cells or according to specific *spreadsheets patterns.*

Proposed Migration Process
Data Model Reverse Engineering

- An heuristic based approach to infer the Data Model was proposed.
- A set of 26 heuristics were considered.
  - 11 heuristics derived from the literature and were adapted to work in this specific context.

Proposed Migration Process
Data Model Reverse Engineering

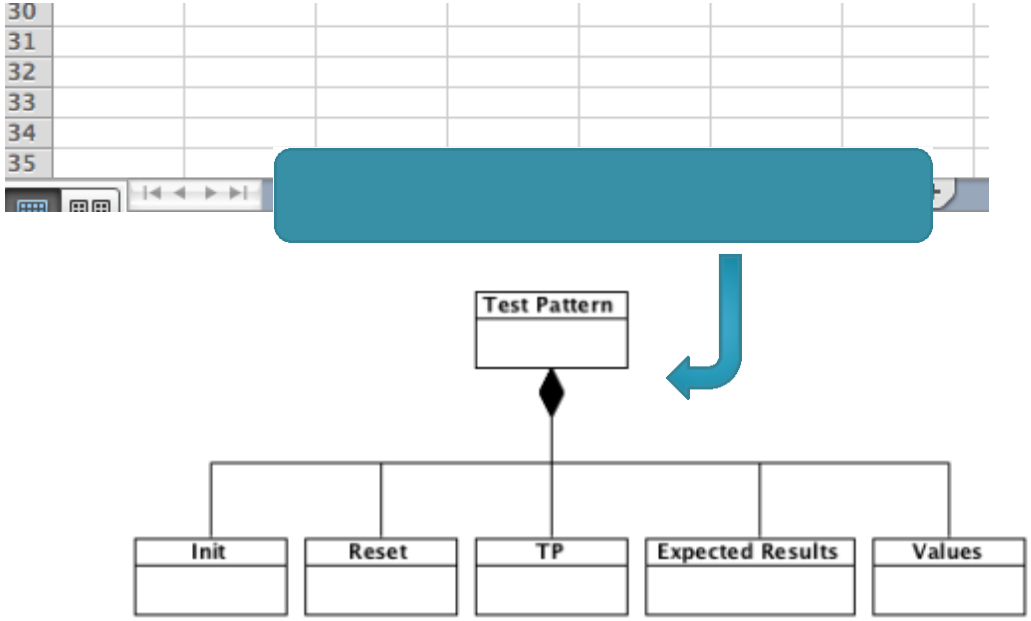
- Heuristics can be grouped in two main classes:
  - Structure based rules (SBRs)
  - Information based rules (IBRs)

Proposed Migration Process
Data Model Reverse Engineering

- Structure based rules (SBRs) analyze the structure and the properties of spreadsheets and their components, such as sheets, cells, cell headers, etc.
  - Used to abstract the set of candidate classes and their relationships;
  - Applied to a single Excel File.

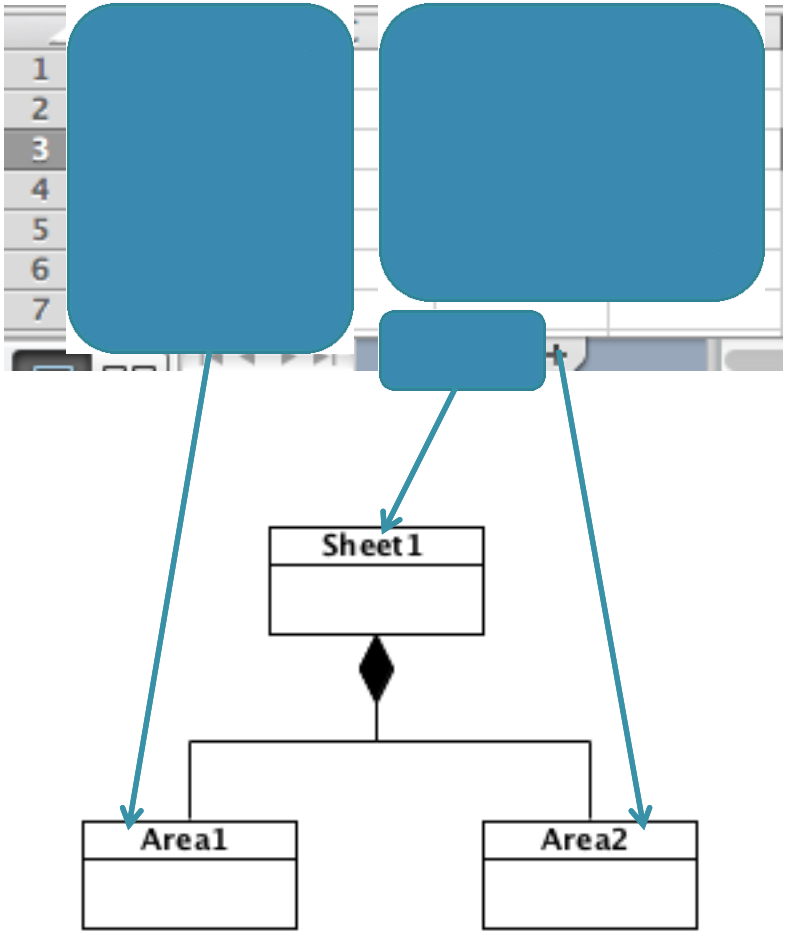
Proposed Migration Process Example of SBR

Rule:
If the spreadsheet contains more than one sheet, **then** it is possible to associate the spreadsheet to a class C and each component sheet to a distinct class S_i, where C has a UML composition relationship with each S_i.



Proposed Migration Process Example of SBR

Rule:
If a sheet S contains sets of consecutive non-empty cells (hereafter *non-empty cell area*) that are well delimited from each other by means of empty cells, **then** it is possible to associate each non-empty cell area to a single class C_i and the sheet S to a candidate class C_S, where S has a UML composition relationship with each C_i.

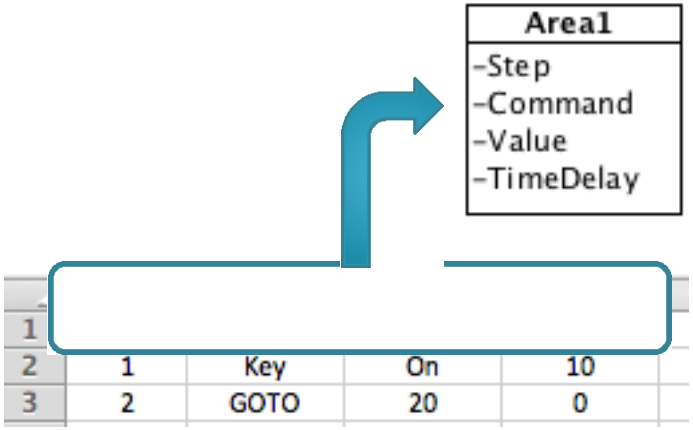


Proposed Migration Process
Data Model Reverse Engineering

- Information based rules (IBRs) analyze the informative content of the cells by looking for repeated data, synonyms, and cells containing well-defined data structures such as array strings, integer matrixes, etc.
  - Used to infer the attributes of classes, the relationships between classes and their cardinalities.
  - Applied to all the Excel Files of the Legacy System.

Proposed Migration Process Example of IBR

Rule:
If the header cells of the columns that discriminated the extraction of a given class A assume the same textual content in all the spreadsheets, **then** these values may be considered attributes of that class.



Proposed Migration Process
Data Model Reverse Engineering

- Selected groups of rules were iteratively applied to the spreadsheets.
- Sets of candidate classes and relationships were automatically proposed.
- Candidates were submitted to domain experts who chose to accept, to refine or to reject them.
- The data model made by 18 classes, 27 relationships, and 95 attributes was reconstructed at the end of the process.

Proposed Migration Process

User Interface and Business Logic Reverse Engineering

- The step was performed to comprehend both the structure and the behavior of the legacy system User Interface (UI) and the functionalities provided by it.
- Program comprehension techniques already proposed in the literature and actually used by professionals in industry were adapted to the VBA code
- Inspection of the VBA source code by both static and dynamic analysis techniques was required due to the lack of documentation.

Proposed Migration Process

User Interface and Business Logic Reverse Engineering

- Static analysis has been used to reconstruct the structure of the UI instances in terms of:
  - component widgets, along with the values assumed by their properties (such as event listeners, event handlers, colors, texts, etc.).

Proposed Migration Process

User Interface and Business Logic Reverse Engineering

- Dynamic analysis techniques were exploited to understand the features provided by the Excel UI.
  - The interaction with the UI and the corresponding code execution were traced in Log files.
  - Log files were post-processed and analyzed in order to:
    - comprehend the VBA functions executed at runtime.
    - reconstruct the business logic provided by the Excel Files.
- The reverse engineered functionalities were validated by the domain experts.

Reengineering

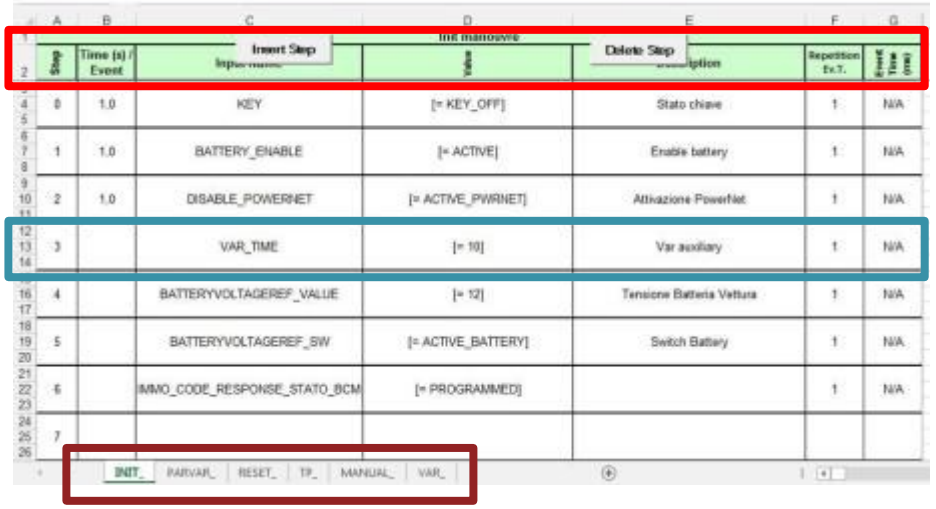
- The models were used to design and develop the new MVC based Information System.
- The system was developed by using the .NET framework and the data persistence was managed by a Microsoft SQL Server DBMS.

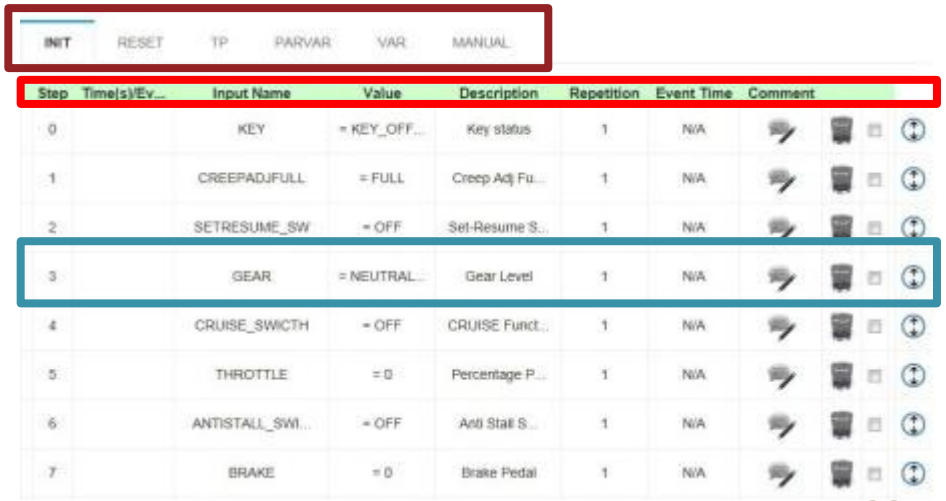
Reengineering

- Additional features of data Import/Export were developed to support
  - the release of the new application;
  - the backward compatibility with existing HIL tool chain.
- Design choices allowed to obtain a new system with no data replication and with an improved maintainability of the stored data.

Reengineering
Analogies between GUIs

Excel GUI





Web Application GUI

Validation

- A study to assess the usability requirements was carried out.
- A group of 8 professional Test Pattern developers was selected.
  - Users of the legacy application and familiar with the new one (attended a 16-hours training course).
- Accomplishment of 25 tasks, using both the legacy and the new Web application, was asked.
  - Selected tasks represented typical Test Patterns editing operations.

Validation

- The usability has been evaluated according to three different metrics:
  - Completion time;
  - Number of editing errors;
  - Number of interactions (i.e., steps) with the user interface.
- The values were collected during the accomplishment of the tasks.

Validation

- Table shows the average results obtained for the three metrics on both the applications.

	AVG #Steps per Task	AVG Completion Time per Task	AVG Editing Errors per task
[redacted]			
Web Application	3.9	7	0
[redacted]			

Validation

- The new application:
  - reduces the overall completion time, the 33% less, on average,
  - reduces the number of steps required for each task the 41% less, on average,
  - avoids any editing error.

- All the Test Patterns edited in the new application have been further translated into Test Objects that were executed with no runtime errors.
  - This prove the effective integration of this application in the HIL testing process.

Conclusions

- A three steps migration project accomplished in an industrial context was presented.
- The cost of the overall migration process:
  - 4 person-months were needed to accomplish the reverse engineering step.
  - 9 person-months for the reengineering step.
  - Costs were considered completely acceptable by the industry.

Conclusions

- The project
  - confirmed the relevance of the spreadsheet-based legacy systems migration in industrial contexts.
  - showed that techniques and approaches of reverse engineering proposed in the literature can be reused with success in migration processes involving systems from industry.
- In future work, we plan to evaluate the reusability of the proposed approach in different industrial contexts involving other spreadsheets-based systems.