



**University of
Zurich** UZH

Department of Informatics

University of Zürich
Department of Informatics
Binzmühlestr. 14
CH-8050 Zürich
Phone. +41 44 635 43 11
Fax +41 44 635 68 09
www.ifi.uzh.ch/dbtg

UZH, Dept. of Informatics, Binzmühlestr. 14, CH-8050 Zürich

Prof. Dr. Michael Böhlen
Professor
Phone +41 44 635 43 33
Fax +41 44 635 68 09
boehlen@ifi.uzh.ch

Zürich, July 22, 2019

MSc Project: Extending a Domain Language for Life Insurance Reporting

With the evolving regulation around the reporting of financial numbers, the management of the associated business data needs to catch up. Excel-based solutions (already previously cumbersome and difficult to maintain) are not really part of the solution-palette anymore, and dedicated software applications come into play.

Most large reinsurance and insurance companies must comply with the new IFRS 17 reporting standards, and as a result such applications must be able to deal with the complex business logic, evolving mathematical methodologies, strict governance rules, legacy systems, and the associated heterogenous data models therein [1]. Furthermore, such applications have to bring together the actuarial and accounting areas in a fashion which is business-oriented (as opposed to having everyone lost in technical details).

Other, older technologies are also revealing to be unsuitable for the job. "One more Warehouse and a few ETL interfaces" does not come close to achieving governance, maintainability, transparency, and business-oriented challenges. Hence, Systemorph has decided to approach the problem by combining various state-of-the-art techniques around data management and reporting – They are:

- A data modeling language in order to define the data model. Such data models are oriented towards business needs, include a clear separation between the various data elements and track the associated dependencies.
- A domain language, which allows for the injection of transparent and easy-to-maintain business-logic. Some examples are:
 - Data validation (which can be applied in the appropriate process step)
 - Real-time transformations during data import



- Mathematical formulas around reporting
- A web-application with a rich RESTful API
- Multi-temporal versioning
- Configurable realistic (e.g. non-linear) workflows

This goal of this project is to build such an application. The context is Life Insurance, and the management of data around mortality tables, which is one of the parameters that serves as input for actuarial calculations. Such calculations are not the focus of this project, but their results are, in the sense that the application should show them in the form of reports.

Furthermore, we try to provide the students with an experience close to one which would occur in “real-life” when developing a proof-of-concept application for a client – in particular, the students will work with:

- Specifications akin to real ones, followed by discussions and question answers, whereby Systemorph will also play the role of the client.
- Azure DevOps tools:
 - Board for tickets
 - Git repository for code
 - SharePoint for files
- Scrum calls
- Unit tests
- Quality Assurance (QA)
- NuGet packages

The tasks undertaken by the students are defined by the deliverable of the project, which is a functional web application covering the:

1. Management of the reference data (e.g. companies, yield curve)
2. Import of generic mortality tables (as available for a certain year and quarter)
3. Management of the parameters required for the calculation of specific mortality tables (e.g. by company, by gender), whereby:
 - (a) Such parameters can be updated on a quarterly basis
 - (b) And calculations should use the latest-available data (e.g. it might happen that for the current quarter there are no updated generic mortality tables, or updated parameters)
4. Calculation of the specific mortality tables, as well as the
5. Export of these specific mortality tables into a separate “actuarial” application (which for the purpose of this project can also be “mockified”)
6. Import of the “actuarial” results (from the “actuarial” application) into the (project) application, including transformations and validations of the data



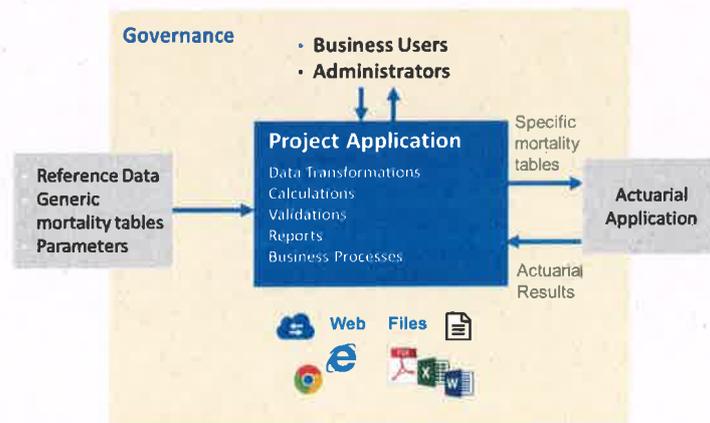
7. Reports (at least 2) for the display of the "actuarial" results, including associated calculations
8. Configuration of two business processes, covering point 1) and points 2-6), respectively, including so-called 'soft' and 'hard' validation rules

Coding will be done in C# and will use functionalities of the Systemorph Vertex platform [2], which is made available as NuGet packages.

The core of the technical work will be around the implementation of:

- An appropriate data model
- An appropriate domain language (or extensions of the domain language elements available from the Vertex platform), which allow provide the transformations and calculations required for points 4-7 above

Schematically, the application functionalities and its interfaces can be depicted as follows:



References

- [1] Systemorph IFRS 17 Compliance Series: <https://systemorph.com/learn/ifrs-17-compliance>
- [2] The Systemorph Vertex Platform: <https://systemorph.com/learn/offering>

Supervisor: Sven Helmer

On Systemorph side: Pedro Fonseca (Head of Business Solutions) and Roland Bürgi (CEO)

Department of Informatics, University of Zurich

Prof. Dr. Michael Böhlen

