Generation of Angular User Interfaces Based on Insurance Configurations*

Nuno Ferreira[1], Manuel Ribeiro[2], Catarina Tomé[3]

i2S Insurance Knowledge
[1] nuno.ferreira@i2s.pt
[2] manuel.ribeiro@i2s.pt
[3] catarina.tome@i2s.pt

Abstract.
The eXtended Visual Group Approach (XVGA) allows to define a process and a software tool to dynamically generate Angular user interfaces based on Insurance Products specification. This approach contributes to optimize the development and maintenance of software products in contexts where information changes rapidly and time to market is a key success factor. The approach defines a meta-model for instantiating the domain specific language (DSL) concepts and a flow of transitions from the server side to the client side to generate multi-device user interfaces based on the DSL. The usability of the generated interfaces was assessed by surveying insurance companies' key users and by comparing the productivity and user satisfaction when compared with previous versions of the user interface.

Keywords: Insurance, User Interface Generation, Angular.

1 Insurance user interfaces

1.1 Problem context

The first i2S insurance applications were deployed on IBM AS400 systems (now IBM iSeries) with black and green static screens, and few usability concerns. The general user interface development guideline was that application users had an expert knowledge of the application and of the business domain (insurance) and so, there was no need to guide them through the application. Nowadays, with the growing number and complexity of systems it is not humanly conceivable that a single user knows all the application or business domain and it is expected that the application provides just the necessary information at a given time and at the same time guides the user to accomplish the required tasks trough the flow of information.

Traditional user interface software tools [1] or declarative models [2] are suitable for developing user interfaces in many devices and solve the problem of generating

* This work has been supported by project ModApNVida (POCI-01-0247-FEDER-009968).

static user interfaces for desktop applications but the changing needs regarding mobility and multi-device deployment at the same time that there is a need to reuse the existing information and business logic in the legacy software systems.

The legacy screens use the business logic and information that resides in the application and this is a starting point to create a new user interface that uses the best practices. There are approaches, like ReGUI [3] that reverse engineer the underlying business logic and create models with the inferred information. This information can then be used to generate new user interfaces. Despite the apparent ease of the depicted process, there are some drawbacks [4] that prevent its out-of-the-box adoption in the insurance legacy systems domain: The generation of a new user interface based on the existing models requires that developers learn a new language to express the user interface specifications; the automatically generated user interface has limitation regarding the usability and fluidity of information between controls that in a traditional (manual) interface generation can be avoided; The inferred interface information that resides on the models is hard to relate with business information and thus it is not always possible to generate a coherent set of controls. Other approaches, like SketchiXML [5] proved themselves as the basis for current mock-up systems with intelligence like UISKEI [6] or Balsamiq Mockups [7] that help the design and assessment of user interface during the software development lifecycle.

Our approach raises the bar one step higher by allowing the analysis and design artifacts being directly translated into controls and screens in the end-user interface by means of configuration of the domain-specific language concepts.

1.2 i2S requirements for generating user interfaces

Since its creation in 1985 by i2S, the insurance software solution has been in constant evolution with the inclusion of new functionalities and new modules, determined by the insurance global market demand and also by the evolution of information technology itself. The development of the insurance solution relies not only on the several (nearly 250) computer technicians and insurance specialists in i2S and other companies representing and implementing the project in other countries, but also on the clients' computer technicians and insurance specialists who adopted this solution (in partnership with i2S technicians). Today, the i2S solution is installed in nearly 50 insurance companies (Life and Non-life), several pension fund management societies and more than 200 agents and brokers, and it is implemented in different countries (Portugal, Brazil, Spain, Poland, Cape Verde, Angola, Mozambique).

Changing a user interface in an application that has more than 30 years (and with some users using it for that long) demands careful analysis and planning. For what requirements are concerned, the new user interface should tackle the difficulties reported by the key users and at the same time ensure that it fulfils the intended requirements and business processes.

Regarding the business users input, we have decided to survey their needs prior to starting the development and measure their productivity and feeling regarding the user interface. This measurement was also conducted with the new interface. The analysis was done using three guidelines: contextual interview, usability and accessibility tests

with key-users and a survey. Two control groups have been established, one from a major Portuguese insurance company and a second from i2S workers. Both control groups average on gender, age and have great expertise on the insurance domain, in the application context and in IT (user-level). The assessment results are detailed in the conclusions.

Regarding the business processes that the solution must address, the chosen solution regards the creation of a dynamic system that is able to automatically create user interfaces based on the AngularJS [5] and Bootstrap [6] frameworks for dynamic web applications. This need has been adressed many times in the current research. Case studies regarding business process reengineering [7] concludes that this is a current need but the techniques most suitable for each case varies and depends on the diagnosis of the situation. Despite that, the need for reverting flows based on paper business processes must rely on a dynamic and configurable system able to be deployed in multiple and different instances, be client-responsive, and independent from manual work (and rework) from IT to generate user-input screens. These requirements gave origin to the eXtended Visual Group Approach (XVGA). This need for process automation is fully aligned with today's paradigms calls for automation in insurance companies. The purpose for insurers is that manual work should only be necessary for exceptions, whereas most insurance companies have typically only been able to automate 30% of key process decisions in the last 30 years [8], which suggests that approaches used so far are not viable to produce the necessary change or provide the needed benefits.

2 The eXtended Visual Group Approach (XVGA)

Customers purchase insurance policies, which are a contract between an insurance company and the insured whereby the insured pays the insurance company a sum of money, called a premium. The premium provides the insured or their beneficiaries the right to be paid an indemnity or receive benefits in the case of certain contractual risks occurring to the insured object or person. An "Insurance Product" is the definition for the operating model of the insurance company and for its IT systems with regards to the contract, it defines the nature of the contract, for example is it Life insurance or automobile (Non-life), what clauses it will include, how they will be priced, which rules will apply over its life cycle. Basically each "insurance product" is a definition for an individual operating model with a high impact on insurance companies because very different kinds of insurance contracts are possible from car insurance to cyber insurance to health and life insurance, and while the basics are the same the data and processes needed to operate insurance contracts resulting from the definition are different.

The information required for a given insurance product must be inputted by the user or gathered from any automatic interface. The problem with the user interface is that it depends on the information configured in the insurance product that varies from product to product, in name, scope, dependency and values. The eXtended Visual Group Approach was defined to give insurance companies the flexibility of inserting values in insurance products without the need of reengineering the software application's screens.

The approach relies, as can be seen on Fig. 1, on a configuration and on a data section. The data section has the insurance product configuration, the subject data, that is, the specific concepts used by the insurance product that are configured. This section relates to the configuration section that comprises the subject-specific information, that is, the relation between data and configuration. This is the basis for mapping and creating groups of information. The information per visual group gives a visual and semantic coherence between subject-specific individual sets of data and enables the creation of visual groups per area. The screen is organized into different areas that depend on the insurance product configuration, required information per section, among others. Finally, there is the visual groups per context. This relates to the application context. We will demonstrate these concepts using i2S Fleet, a new Non-Life pilot component of i2S' offering that uses all the technologies researched in the context of POCI-01-0247-FEDER-009968 project.

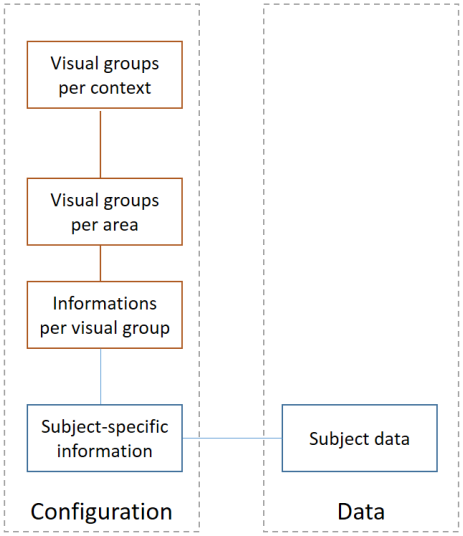

Fig. 1. XVGA meta-context

In the "Configuration" section we must define all the configuration-related issues that impact the visual groups for a given context (application), screen areas and inside those, the visual groups (controls grouped together that form a single unit of work).
The "Subject Data" relates to the information that characterizes the visual group context and all the information that must be displayed for assuring the user input and application output, grouped by areas and visual groups.
The "Visual groups per context" defines and describes existing visual groups and areas, and common configuration settings shared among the areas, visual groups and controls.
The "Visual groups per area" identifies and groups all the visual groups for a given area.

The “Information per visual group” relates to the meta-information for positioning the visual group on the final layout. It includes name, level, visual group, description, sequence, tooltip, among others.
The “Subject-specific information” regards the meta-information for the field itself and includes its name, visual group information, type of field, row and column, tooltips, level, and others.
The “Data” section comprises the instances of the data used by the control defined in the subject-specific information.
The implementation of this approach is in Angular, the webservice layer (REST) uses Apache CFX and the database access is done using JPA. Other additional constraints include the fixed date format that will be subject to change in future releases.
The extended visual group approach also supports defining fields for sums, totals, relationships, specific controls (date time, combo, and selection) and others that contribute for a full-fledged responsive user interface.

3 Use-case in Fleet management software

The XVGA approach was demonstrated using a real industry scenario, in the development of a prototype tested with anonymized end-user data. The software is called i2S Fleet and it is a product whose insured object is composed of motor vehicles. The fleet product family is characterized by the large number of vehicles it comprises, and which aims to respond to the specific management needs of this type of insurance. The main characteristic of a fleet insurance policy is that it requires a heavy management burden, as is the case with products that include a large number of insured objects. Therefore, the Fleet product must be considered in the context of business customization. That is, each fleet, in addition to having multiple insured objects, may have distinct characteristics, which forces the company to adjust to the needs of each client and, therefore, put in a greater management effort.

Due to the previous, the need to automatically generate an interface earns provides earnings in productivity and eases the burden of management effort required to manage all the variability of the fleet insured products. As already mentioned, the fleet insurance is characterized by insuring a group of vehicles. This group of vehicles may be divided into smaller subgroups, which are created according to homogeneous attributes, and which allow the definition of distinct contract conditions, coverages and commercial discounts per group. With this tool, the fleet management tasks do not require as much work, as all vehicles within the same subgroup are handled in the same way, and often automatically (for instance, when adding a new vehicle, it will integrate the subgroup of vehicles of the same type within the fleet, and automatically assume the conditions of that subgroup).

The required configuration to support this variability is depicted in **Fig. 2.** Fleet visual group configuration. In this configuration schema, the subject data encompasses a set of business objects, like policy information, fleet group information, that support the context for the variability associated with the fleet insurance product. There are also

objects that support the system itself, like monetary information's, "objects" by themselves, among others. These are related to the generator's configuration schema specific for this software product.

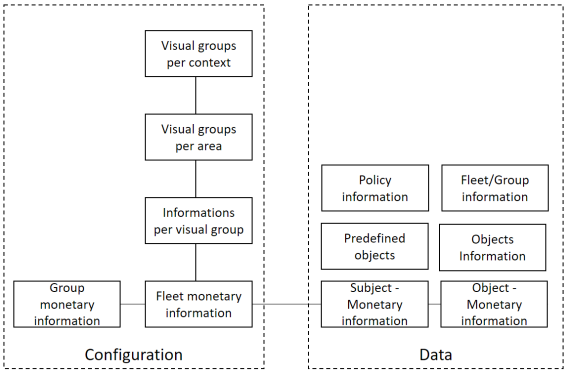

Fig. 2. Fleet visual group configuration

The visual field definitions are translated into language primitives, in this case, angular (typescript) language constructors, and injected into the processor for displaying a responsive screen.

4 Conclusions

Our approach was assessed by a selected group of end-users, where the configurations where made to assure compliance with the elicited requirements. The followed methodology encompassed three major stages:

1. Contextual interview, where end-users discussed a set of topics regarding the generated test user interface;
2. Usability and Accessibility tests, where the end-users used the XVGA approach to configure the system and check the generated interface regarding usability and common controls access and;
3. Survey, where the end-users formally expressed their opinion on the process and the result.

The test was conducted in a controlled environment, with two control groups. The first group with members from an insurance company (i2S customer), business specialists from the Fleet management group and IT users. All males with ages ranging from 30 to 49 years old. The second group of only i2S workers, with entry-level knowledge of insurance and IT. Males and females with ages between 30 and 60.

Major findings are that the developed approach is suitable for the needs of both groups and that the process reduces time and complexity of launching a new interface, and thus enabling faster time to market and increasing productivity of the insurance company. Some weaknesses were also detected and subject to improvements in later versions of the XVGA approach.

This paper is framed within project POCI-01-0247-FEDER-009968 which has the purpose of conceiving a Non-Life insurance platform able to be executed executable in a multi-device and cloud environment, thus making it scalable, accessible and adjustable to different application contexts as well as to intelligent risk management. The initiative was supported by Portugal 2020, through Programa Operacional Competitividade e Internacionalização in the scope of Sistema de Incentivos à Investigação e Desenvolvimento Tecnológico, with funding from Fundo Europeu de Desenvolvimento Regional (FEDER). The elegible investment was of 1,176,414.68€, corresponding to a FEDER contribution of 848,506.39€

References

[1] S. E. H. R. P. Brad Myers, "Past, present, and future of user interface software tools," em *ACM Transactions on Computer-Human Interaction (TOCHI)*, 2000.

[2] P. P. D. Silva, "User interface declarative models and development environments: A survey," em *International Workshop on Design, Specification, and Verification of Interactive Systems*, 2000.

[3] A. P. J. P. F. Inês Coimbra Morgado, "Dynamic reverse engineering of graphical user interfaces," *Int. Journal on Advances in Software,* pp. 224-246, 2012.

[4] J. Vanderdonckt, "Model-driven engineering of user interfaces: Promises, successes, failures, and challenges," em *Proceedings of ROCHI*, 2009.

[5] [Online]. Available: https://angularjs.org/.

[6] [Online]. Available: http://getbootstrap.com/.

[7] D. Grant, "Business analysis techniques in business reengineering.," *Business Process Management Journal,* vol. 22, nº 1, pp. 75-88, 2016.

[8] B. &. Company, "Global Digital Insurance Benchmarking Report. Pathways to success in a digital world," Bain & Company, 2015.