Model-Based Simulation of Legal Requirements: Experience from Tax Policy Simulation

Ghanem Soltana, Mehrdad Sabetzadeh, and Lionel C. Briand
SnT Centre for Security, Reliability and Trust
University of Luxembourg, Luxembourg
{ghanem.soltana, mehrdad.sabetzadeh, lionel.briand}@uni.lu

***Abstract*—Using models for expressing legal requirements is now commonplace in Requirements Engineering. Models of legal requirements, on the one hand, facilitate communication between software engineers and legal experts, and on the other hand, provide a basis for systematic and automated analysis. The most prevalent application of legal requirements models is for checking the compliance of software systems with laws and regulations. In this experience paper, we explore a complementary application of legal requirements models, namely *simulation*. We observe that, in domains such as taxation, the same models that underlie legal compliance analysis bring important added value by enabling simulation. Concretely, this paper reports on the model-based simulation of selected legal requirements (policies) derived from Luxembourg's Income Tax Law. The simulation scenario considered in the paper is aimed at analyzing the impact of a current tax law reform proposal in Luxembourg. We describe our approach for simulation along with empirical results demonstrating the feasibility and accuracy of the approach. We further present lessons learned from the experience.**

***Index Terms*—Legal Requirements, Modeling, Simulation.**

I. Introduction

Software systems are increasingly subject to laws and regulations. To develop a legally compliant system, software engineers need to interpret, typically in collaboration with legal experts, the relevant legal texts and derive from these texts legal requirements that the system under development must fulfill. A common way for representing legal requirements is through *modeling*. Models offer intuitive means for communication between software engineers and legal experts, thus facilitating building a shared understanding of the legal requirements. Models further provide a useful basis for systematic and automated analysis of legal compliance.

Several strands of work employ models for elaborating legal requirements and assessing whether and to what extent these requirements are met by a given system. These strands include the large body of research concerned with the application of goal models to laws and regulations, e.g., [1], [2], as well as a number of conceptual modeling techniques aimed at representing the semantic metadata of legal texts, such as key legal abstractions and modalities, e.g., [3], [4], and the structural makeup of legal texts, e.g., [5]–[7].

In recent years, we have been exploring the application of modeling for automated analysis of compliance to the tax law. The tax law, along with several other categories of law such as social benefits and customs duty laws, are highly operationalized. These laws typically structure the compliance requirements into detailed *policies*. Public administration IT systems need to implement and comply with these policies.

In our earlier work [8], we proposed a UML-based approach for expressing legal policies. An observation that we made in this context is that the resulting policy models have added analytical value beyond what we initially had in mind, which was to use the models for compliance testing of government IT systems. An interesting additional application of the policy models is for *simulation*. The goal of simulation is to predict the impact of a set of proposed changes to laws and regulations, and ensure that the proposed changes will bring about the desired outcomes without unwanted side effects. This new application of policy models prompted us to develop a model-based simulator [9], drawing on the same modeling approach developed previously for legal compliance testing.

In this paper, we report on a case study where we apply, for the first time, our simulator for analyzing the impact of a *real* legal change proposal. The change proposal concerns the "joint taxation" policy in Luxembourg's Income Tax Law. This policy, which is also known as "income splitting" in some other countries, enables spouses to attribute, under certain conditions, some of the income of the higher-earning spouse to the lower- (or non-)earning spouse. The policy often leads to a reduced overall tax obligation for the spouses. The Government of Luxembourg is currently considering a proposal to abolish (repeal) joint taxation. Our case study employs simulation in order to examine how this potential reform is likely to impact personal income taxes. The case study is motivated by the following Research Questions (RQs):

RQ1: Can we bring together and model all the information necessary for performing a real-world simulation scenario? While the feasibility of building individual policy models was examined in our prior work [8], [9], we did not previously investigate whether one can build a seamless set of models that need to be considered together in a real-world simulation scenario. Furthermore, our simulator includes a probabilistic data generator to create artificial data in situations where access to real simulation data is restricted (the situation we have to deal with in our case study), or where real data is unavailable, e.g., when we are simulating a new policy for which no historical data exists. RQ1 aims to study the feasibility of building a coherent set of policy models for joint taxation and other closely-related policies. This RQ further

looks into whether we can instrument our data generator with sufficient probabilistic guidance to generate data for simulating joint taxation.

RQ2: Are the simulation results credible? RQ2 is aimed at comparing the results produced by our simulator against publicly-available statistics on tax contributions from taxpayers in different income brackets. Indirectly, RQ2 develops confidence about two important factors: First, is the level of abstraction at which we express the policy models a good fit for our analytical purpose? And second, does the artificial data generated by our simulator serve as a good replacement for real data, i.e., taxpayer records, to which we have no access? Our answer to RQ2 will be based on comparing the results of simulating the status quo, i.e., the situation where joint taxation is enforced, against *current* tax contribution statistics. The analysis that we performed in reality for our public service partners was to compare the status quo against a potential future where joint taxation has been abolished [10]. Using our simulator for enacting this scenario does not add new conceptual element to our framework; to do so, we simply provided a current (i.e., with joint taxation) and a future (i.e., without joint taxation) set of policies, subjected the two sets to the same simulation data, and quantified the difference between the simulation results. Since we naturally do not know at this stage how accurate our predictions are, we rely on the status quo for analyzing the credibility of the simulation results.

In addition to addressing the above two RQs, we summarize in this paper the lessons that we learned from our experience. The lessons cover a number of important considerations in relation to traceability between models and legal texts, and making policy models easier to understand for legal experts.

We believe that our work in this paper is useful to the Requirements Engineering community in three related ways: First, we demonstrate how models of legal requirements built for enabling compliance analysis can further be exploited, with minor adaptations, for another important analytical purpose: simulation. Modeling in general, and requirements modeling in particular, require upfront investment when adopted in industrial settings. Being able to use the same models for multiple purposes makes modeling more cost-effective, in turn contributing to the wider adoption of models in industry. Second, one can use simulation as a vehicle for validating legal requirements models by comparing the results of simulation against expectations. Since, in the context of our work, compliance analysis and simulation build on the same models, validation via simulation contributes to building more accurate models for compliance analysis. Finally, while our RQs and lessons learned are naturally oriented around simulation due to the nature of our case study, we anticipate our conclusions and observations, e.g., about modeling effort and the accuracy of analysis results, to generalize to a large extent to compliance analysis.

Structure of the Paper. Section II outlines our simulation framework and tool support. Section III describes the design of our case study. Section IV presents the case study results and answers our motivating RQs, stated earlier. Section V reflects on the lessons learned. Section VI compares with related work and Section VII concludes the paper.

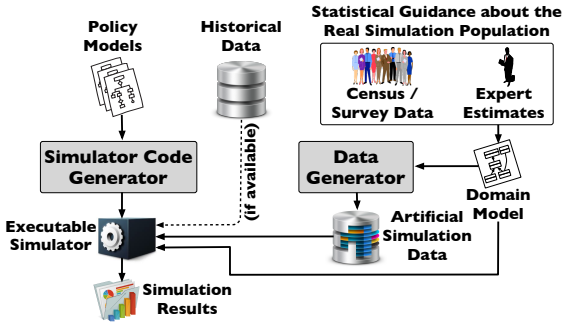

Fig. 1. Overview of our Simulation Approach

II. Approach

Fig. 1 presents an overview of our simulation approach. The approach takes as input a set of policy models that need to be simulated. Policy models are interpretations of the legal texts and provisions that are relevant to the intended simulation activity. For the taxation domain, the policies are concerned primarily with the calculation of quantities such as credits, deductions and taxes. We discuss and exemplify in Section II-A the notation that we use for specifying policy models. Although not shown in Fig. 1, all policy models need to be validated by legal experts before simulation. The goal of the validation is to ensure that the models are a faithful representation of the relevant legal requirements. We describe our validation process in Section III-B.

The input policy models are automatically transformed into an executable simulator. This process is transparent to the user, thus eliminating the need to manually review or manipulate any software code. The data to be processed by the simulator can be provided through two alternate means: If historical data, e.g., a database of tax records, is available, this data can be fed directly to the simulator for processing. Alternatively, if historical data is unavailable, inaccessible, or incomplete, our approach automatically generates artificial data for simulation.

The basis for automatic data generation is a domain model. The main purpose of a domain model in our approach is to enable guiding the data generator towards creating realistic data, i.e., data whose characteristics are as close as possible to those of the population being simulated. Specifically, we use the domain model as a container for providing statistical guidance to the data generator. This guidance is typically derived from existing census and survey statistics. Expert estimates may be further utilized for exploring future (uncertain) contingencies, or for addressing possible gaps in the statistics. We describe in Section II-B how the statistical guidance is expressed.

Once the simulator has processed the simulation data (either historical or artificial), the results are aggregated into spreadsheets and charts and provided to the user. Our approach further supports simulation result differencing. This means that the user can provide an original and a revised set of policy

TABLE I
GLOSSARY FOR THE INPUTS TO THE POLICY MODEL OF FIG. 2

Input name	Description
is_taxed_jointly	Yes if a given taxpayer is taxed jointly with another taxpayer; otherwise, no. The value of this input is determined via the application of another policy model: joint taxation.
is_married	Yes if a given taxpayer is married; otherwise, no.
are_both_spouses_ non_resident	Yes if a given couple are both non-residents; otherwise, no.
is_living_separately	Yes if a given taxpayer is "de facto" separated (*séparation de fait* in French); otherwise, no.
is_divorced	Yes if a given taxpayer is divorced by mutual agreement; otherwise, no.
is_divorced_by_ court_order	Yes if a given taxpayer is divorced by court order; otherwise, no.
has_separation_ transition_state	Yes if a given taxpayer is in a transition state after separation; otherwise, no. The transition state is granted when a taxpayer's date of separation is within the past three years.
is_widower	Yes if a given taxpayer is a widower; otherwise, no.
has_widower_ transition_state	Yes if a given taxpayer is in a transition state due to having been widowed within the past three years; otherwise, no.
is_receiving_ allowance_ for_children	Yes if a given taxpayer is receiving some benefit or allowance for their children; otherwise, no. Examples of allowances include childbirth benefit and child allowance.
taxpayer_age	Age of a given taxpayer.
is_couple_living_ apart	Yes if a given couple do not live at the same address; otherwise, no.
local_professional_ household_income	Sum, for a given household, of the professional incomes taxed in Luxembourg. Categories of professional income are defined by *Articles 14, 61, 91 and 95* of *LITL*.
total_professional_ household_income	Sum, for a given household, of all professional incomes.

models, simulate both sets, and compare the simulation results to quantify the impact.

In this paper, we consider our simulation approach exclusively from the perspective of users. We therefore address only the inputs to and the outputs from our simulation approach, rather than the technical machinery behind the approach. A detailed treatment of the technical components of the approach is available in our previous work [9]. Below, we present the modeling aspects of our approach and outline tool support.

A. Policy Models

Policy models capture the workflow for realizing a given policy using a customized form of UML activity diagrams [8]. To illustrate, Fig. 2 shows a policy model, named Tax Class Categorization (TCC), whose function is to assign a tax class to a taxpayer, as per the provisions of Luxembourg's Income Tax Law (*LITL*). This policy is one of the policies involved in our case study (Section III). *LITL* defines three tax classes: tax class 1, tax class 1.a, and tax class 2. The tax class determines the taxation rate and formula to apply for calculating taxes due. For example, the lowest tax rates are used for taxpayers belonging to tax class 2. A taxpayer is assigned a tax class based on their personal situation, including among other factors, family and residence status.

To read a policy model, one can first explore the inputs associated with the policy. The inputs are listed on a policy model's left side. For example, the input *is_married* in the policy model of Fig. 2 is "yes" for a given taxpayer if and only if the taxpayer is married. We formalize the inputs to a policy model using expressions written in the Object Constraint Language (OCL) [11]. For example, the OCL expression for *is_married* is: **self**.isMarried(Constants.TAX_YEAR), where **self** refers to the taxpayer being simulated and Constants.TAX_YEAR is the current taxation year, represented as a static attribute of a class named Constants. The OCL expressions for the inputs are not shown in the policy model of Fig. 2 (see [8], [9] for examples of policy models with their inputs expressed in OCL).

As we mentioned earlier, all policy models, including their inputs, have to be validated by legal experts. However, we have observed that legal experts often find it difficult to work directly with formal languages such as OCL [8]. To tackle this issue, we complement the OCL definitions of the inputs by a glossary. The glossary contains, for each input, an intuitive description written in natural language. Table I shows the glossary describing the inputs to the policy model of Fig. 2.

Once familiar with the inputs to a policy model, one can proceed to explore the workflow of a policy from the initial node, denoted by a circle, and follow the different paths leading to the update function(s), denoted by an *«update»* stereotype. An update function is an action performed within the workflow of a policy in order to record the outcome resulting from the simulation of that policy (see [9] for a complete list of the stereotypes in our customized activity diagram notation).

To illustrate, we describe the workflow of the TCC policy model in Fig. 2. The first criterion to consider when assigning a tax class to a taxpayer is whether the taxpayer is taxed jointly with a spouse. This criterion is captured by the diamond-shaped decision node. Depending on the value of the input *is_taxed_jointly*, a suitable path (yes or no) is taken out of the decision node. A taxpayer who is taxed jointly is assigned tax class 2. If a taxpayer is not taxed jointly, then other criteria, including marital status, will determine tax class 2 eligibility. For instance, a widower taxpayer belongs to tax class 2 if they have lost their spouse in the past three years (*has_widower_transition_state*). Similarly, a taxpayer who has had a divorce in the past three years belongs to tax class 2. Finally, married non-resident taxpayers who are living at the same address belong to tax class 2, provided that they realize more than half of their professional income in Luxembourg.

Taxpayers who do not belong to tax class 2 might belong to tax class 1.a. Specifically, tax class 1.a covers: (1) taxpayers who are receiving some child allowance, (2) taxpayers who are aged at least 64, (3) widowers who have lost their spouse prior to the last three years, and (4) non-resident married taxpayers who are not taxed jointly, but who are living at the same address and realizing less than half of their professional income in Luxembourg. Taxpayers who do not belong to either tax class 2 or tax class 1.a are assigned tax class 1.

B. Domain Model

We capture a domain model using UML class diagrams. As we noted earlier, the main role of a domain model in our approach is as a vehicle for providing statistical guidance to

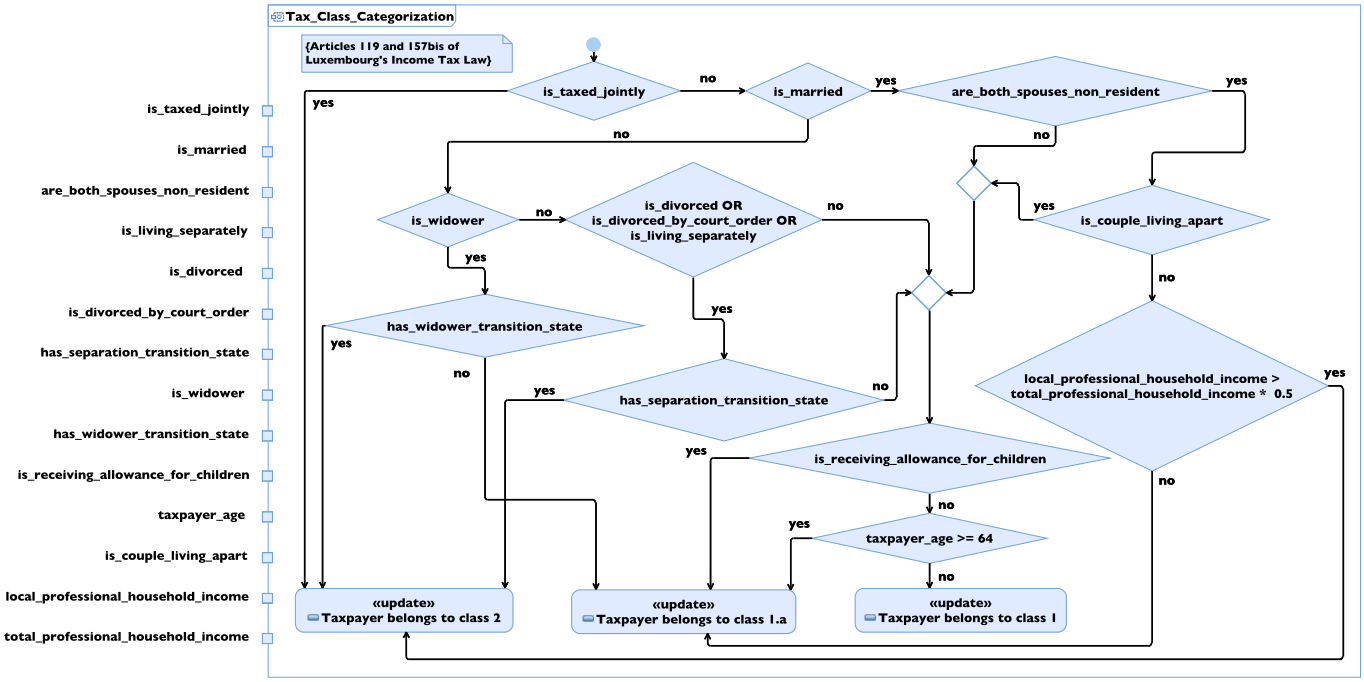

Fig. 2. Policy Model for Tax Class Categorization (TCC)

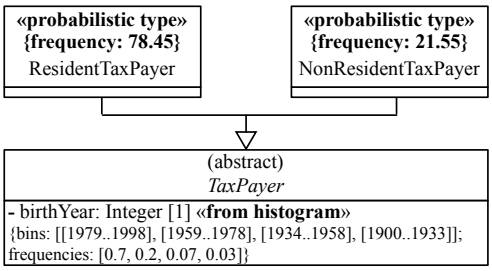

Fig. 3. Domain Model Fragment Extended with Statistical Guidance

our data generator. We follow standard practices for domain modeling [12] and thus do not elaborate the domain model construction process in this paper. To express statistical guidance over a domain model, we extend UML class diagrams with explicit probabilistic notions, including relative frequencies, distributions, and conditional probabilities. We have formalized these probabilistic notions as a UML profile [9].

Fig. 3 illustrates the application of some of the above probabilistic notions on a small fragment of the domain model built as part of our case study. The *«probabilistic type»* stereotypes applied to the specializations of the *TaxPayer* class state that ≈78% of the taxpayers are resident and the remainder are non-resident. The *«from histogram»* stereotype attached to the *birthYear* attribute provides, via a histogram, the birth year (age) distribution of the taxpayers. Technical details and further examples on how we express and use such statistical guidance is given in our previous work [9].

C. Tool Support

We have developed a prototype tool that covers all the steps in our simulation approach. Building the policy models and the domain model, and providing statistical guidance for simulation data generation can be done in any EMF-based modeling environment (eclipse.org/modeling/emf/) with support for UML profiles [13]. We have used Papyrus (eclipse.org/papyrus/) as our modeling environment. The executable simulator is generated via model-to-text transformation. This transformation is implemented in Acceleo (eclipse.org/acceleo/). Our simulation data generator has been developed as a plugin for Eclipse (eclipse.org). Our model-to-text transformation and data generation algorithms have been documented in our previous work [9], [14].

III. Case Study Design

In this section, we report on a case study where we apply the simulation approach presented in Section II for analyzing the impact of the (potential) abolishment of the joint taxation policy in Luxembourg. Joint taxation is done by considering spouses who fulfill certain eligibility criteria as a single tax unit. The effect is that the difference between the income of the higher-earning spouse and the lower- (or non-)earning spouse is split between the two, thus potentially giving the higher-earner a lower tax rate and the lower-earner a higher tax rate. In many circumstances, the policy yields a lower overall tax obligation for the spouses. The case study was conducted in collaboration with the Government of Luxembourg.

A. Case Selection

Our main criterion for case selection was to work on a real and practically-relevant simulation scenario. As long as this criterion was met, our case selection strategy was *opportunistic*. Once the abolishment of joint taxation was suggested to us as a possible case study, we conducted a preliminary investigation to ensure that the case study had

a realistic chance of being brought to completion. To this end, we considered three factors: (1) whether the (at this stage, ballpark) level of modeling effort is reasonable given the research team's resource constraints, (2) whether we have access to legal experts who can validate the models built throughout the case study, and (3) knowing a priori that we will not have access to real tax data, whether there are publicly-available sources where we are likely to find relevant statistical information for guiding the data generation process. We started conducting the case study after initial positive indications about the factors above.

B. Data Collection Procedure

Data collection was targeted at: (1) building policy models and a domain model for the case study, and (2) gathering the relevant probabilistic information and incorporating this information into the domain model.

Policy and domain model construction. The legal experts in our case study directed us to the legal provisions (segments of legal texts) relevant to the intended simulation activity and informally described the interpretation of these provisions to us. Subsequently, we followed our proposed modeling methodology [8] for modeling the policies and the domain.

To validate the developed models, we held regular walk-through sessions with the legal experts. The experts had already received training on how to read and understand simple activity and class diagrams [8]. For each policy model, the legal experts would first review the glossary definitions for the inputs to that model (see Section II-A). The experts would then go through the workflow of the policy model and provide feedback. The validation of the domain model was intertwined with that of the policy models. Specifically, for each policy model, we discussed with the experts the pertinent classes and associations of the domain model and obtained feedback. The validation process was iterative. We kept refining and (re-)validating the models until the legal experts deemed the models to be complete and precise representations of the underlying legal provisions.

Extending the domain model with statistical information. Once the domain model had sufficiently stabilized, we started annotating its elements (classes, attributes, associations) with statistical information. Specifically, for each element in the domain model, we did the following: First, we examined the objective data sources that were available to us, notably public census data and statistics published by the Government, for a suitable distribution. If a suitable distribution could not be found, we asked the experts for estimates and suggestions. In addition, we captured in our domain model any dependencies we knew of between the elements, either from the objective sources or from the experts' domain knowledge. An example dependency is the following: "If a taxpayer has a pension income, the taxpayer must be older than 57" (dependency between income type and age). We capture such dependencies in our domain model as conditional probabilities. Conditional probabilities provide more flexibility than deterministic rules for dealing with dependencies. For the dependency above, we could, for example, state using conditional probabilities that "If a taxpayer has a pension income, then the taxpayer's age should be between 57 and 65 with a probability of 90% and should be greater than 65 with a probability of 10%". A complete list of the statistical information in our case study is provided in Section IV.

C. Analysis Procedure

Analyzing the feasibility of modeling. To analyze whether our simulation approach has a realistic chance of being used in practice, we kept track of the level of modeling effort required by our approach. We further monitored for potential situations where our modeling notation did not provide adequate expressive power. Finally, we examined whether we could enrich our domain model with sufficient statistical information for guiding simulation data generation.

Analyzing the quality of simulation input data. Before using an automatically-generated data sample for simulation, we needed to ensure that the sample is of reasonable quality. We did so by performing sanity checks that compared the statistical distributions of the sample against those of the real taxpayers population, i.e., the distributions attached to the domain model. An example sanity check is to verify whether the relative distribution of resident versus non-resident taxpayers in a generated sample indeed matches what we know about the real population in Luxembourg.

All the statistical distributions in our case study are histograms. For sanity checking of the generated data samples, we thus needed a metric for comparing histograms. We chose the (normalized) Euclidean distance [15] for this purpose. This metric, whose value is between 0 and 1, measures how far two histograms are from one another. For a given pair of histograms, a Euclidean distance close to 0 means that the histograms are well-aligned; whereas a Euclidean distance close to 1 means that the histograms differ significantly.

For example, suppose that the distribution of taxpayers' residential status in the generated simulation data is: 90% resident versus 10% non-resident. As we showed earlier in the domain model fragment of Fig. 3, the actual proportions for the real population are ≈78% resident versus ≈22% non-resident. When these two distributions are represented as histograms, the Euclidean distance between them is ≈0.2.

We deem the sanity check for a specific distribution derived from an automatically-generated data sample as being successful if the Euclidean distance between that distribution and the corresponding actual distribution (attached to the domain model) is sufficiently small. We use a failure threshold of 0.1 to decide whether a sanity check succeeds or fails. Checks that yield a distance below the threshold succeed; checks that yield a distance above the threshold fail. If a generated sample failed any of the sanity checks, we repeated data generation with a larger sample size. We increased the sample size until all sanity checks passed.

Analyzing the credibility of simulation results. After completing a simulation run, we verified the credibility of the simulation results by comparing the results against the aggregate

information we had about the tax contributions of households in different income brackets. Similar to the sanity checks performed on the simulation input data, we used the Euclidean distance metric for determining how close the simulation results were to reality. As we noted earlier in Section I when presenting RQ2, the basis for assessing simulation credibility is the status quo. In other words, we do not assess the credibility of our prediction about the impact of abolishing joint taxation, since, naturally, no data exists about the impact of this potential future change. This said, achieving reasonable accuracy in simulating the status quo is an indication of the adequacy of our modeling methodology and automated data generation process. This in turns provides confidence about the credibility of the predictions made using our approach.

IV. Results and Discussion

The simulation of joint taxation involved the construction of three policy models and a domain model. The policy models are: (1) *Joint Taxation (JT)* to determine whether a taxpayer is eligible for joint taxation, (2) *Tax Class Categorization (TCC)*, shown in Fig. 2, to determine a taxpayer's tax class, and (3) *Extra-Professional Deduction (EPD)*, to grant, under certain conditions, a special deduction to a taxpayer who is taxed jointly. JT, TCC and EPD respectively have, 111, 93, and 73 elements, where an element can be an input, output, decision, action, flow, intermediate variable, expansion region, or a constraint. The domain model has 16 classes, 4 enumerations, 6 associations, 11 generalizations, and 24 attributes.

In addition to the models above, our case study used a pre-existing implementation of the formulas for calculating taxes due. These formulas could have been captured as policy models. Nevertheless, since our case study did not involve any modifications to these formulas, we elected to use the available implementation. This decision was motivated by avoiding the need to re-validate the formulas, which had been already extensively tested.

A. RQ1: Can we bring together and model all the information necessary for performing a real-world simulation scenario?

We consider three aspects for answering RQ1: (1) whether the modeling effort is practical, (2) whether our modeling notation is expressive enough, and (3) whether we could successfully provide the input (statistical information) necessary for the generation of simulation data.

With regard to the modeling effort, and as noted in Section III-C, we kept track of the effort spent on constructing the models (from scratch) and validating them. Table II presents the effort for model construction and validation. The effort for constructing the models is inclusive of the time spent on reading the relevant legal texts and preparing glossaries (exemplified in Table I) for the policy model inputs. Modeling the policies relevant to joint taxation (JT, TCC, EPD) took 15 person-hours (ph). Building the domain model took 7 ph, including the effort for annotating the domain model with the statistical guidance for data generation (discussed later). The model construction activities were led by the first author who has 4 years of experience in Model-Driven Engineering. The validation of the models with legal experts took 5 ph in total, of which 3 ph was spent on the policy models and the remaining 2 ph on the domain model. The above results suggest that the modeling effort for our case study was practical.

TABLE II
Effort for Building and Validating the Models

Activity	**Model**			
	JT	TCC	EPD	Domain Model
Model construction (person hours)	7	5	3	7
Model validation (person hours)	1.5	1	0.5	2

We note that to simulate the impact of a set of proposed changes to laws and regulations, we need to capture both the original and the modified sets of policies. In our case study, we spent negligible effort on creating the modified set of policies because of the nature of the underlying legal change. Specifically, abolishing joint taxation entailed: (1) dropping the JT and EPD policy models from the modified set (thus leaving only TCC in the modified set); and (2) a slight modification to TCC in order to link the initial node to the *is_married* decision thus bypassing the *is_taxed_jointly* decision in the original model (see Fig. 2). When changes to laws and regulations involve more extensive alterations, e.g., adding new policies, the effort for building the modified set of policy models will be proportional to the extent of the alterations.

The second aspect of RQ1 has to do with the expressiveness of our modeling notation. In our case study, we faced only one situation where an extension to our notation was necessary. More precisely, our original notation [8] defines three possible sources from which an input to a policy model can originate: (1) *records*, e.g., for taxpayers' incomes and expenses; (2) *legal texts*, e.g., for monetary values explicitly written into the text of legal provisions, and (3) *agents*, for any input that needs to be supplied by a human, e.g., an eligibility criterion that should be ascertained by a tax officer.

The above three sources do not consider the situation where an input to a given policy model may correspond to the output from another policy model. For example, the *is_taxed_jointly* input of TCC in Fig. 2 is in fact the output of JT. If such relationships are not explicitly modeled, then the simulator cannot determine a correct order for executing the policy models. To address this problem, without having to require users to define an execution order for the policy models, we added a fourth input source, namely *policy*, to state that the value of a certain input to a policy model is determined through the execution of another policy model. This additional construct enables the simulator to automatically find an appropriate execution order for the policy model and warn the users if any cyclic relationships between the models are detected. Aside from this minor change to our policy modeling notation, we found the notation to be expressive enough for our case study.

The third and final aspect of RQ1 concerns whether we could find the statistical information necessary for running our data generator and instantiating our domain model. We

TABLE III
STATISTICS FROM PUBLIC CENSUS DATA AND GOVERNMENTAL SOURCES

	Statistic	Description
1	*Residence status*	Relative distribution of resident versus non-resident taxpayers (see Fig. 3).
2	*Age*	Distribution of the population by age (see Fig. 3).
3	*Household size*	Distribution of the size of households, e.g., 63% of households are composed by more than two persons.
4	*Types of civil union*	Relative distribution of civil unions, e.g., 95.6% of the unions are marriages.
5	*Income types*	Relative distribution of different income types, e.g., 1.4% of incomes are of type agriculture.
6	*Income amounts*	Distribution of gross annual income for households.
7	*Workers per household*	Distributions of households based on the number of active workers in the household.
8	*Divorce rate*	The percentage (by year) of individuals whose marital status has changed from married to divorced.
9	*Divorce types*	Relative distribution of divorce types, e.g., by mutual agreement.
10	*Widower rate*	Percentage of the population that has been widowed in a given year (past 10 years).
11	*Income for pensioners*	Specialization of income distribution for pensioners.
12	*Income for traders*	Specialization of income distribution for traders.
13	*Age of pensioners*	Specialization of age distribution for pensioners.
14	*Foreign income types*	Specialization of the distribution of income types for non-resident taxpayers.
15	*Residence status based on spouse's residence status*	Specialization of the distribution of resident versus non-resident taxpayers based on the residence status of their spouses.

consulted two information sources for statistics about the population of taxpayers in Luxembourg: (1) census data from STATEC (statec.lu/), and (2) a recent report published by Luxembourg's Ministry of Finance [16]. We extracted from these sources statistics about 15 quantities relevant to our simulation scenario. These statistics are described in Table III.

Rows 1 to 10 in Table III are general statistics about the entire population, and Rows 11 to 15 are specialized statistics that replace some of the general ones under certain circumstances. For example, pension incomes require special treatment and are handled differently than, say, employment incomes. This is because pensions are subject to certain regulations that constrain their amounts to a predefined range. During simulation data generation, if a taxpayer has, for example, an employment income, then the general *Income amounts* statistic is used for (probabilistically) generating an income amount; whereas, if a taxpayer is receiving a pension, the specialized statistic for pensions, i.e., *Income for pensioners*, is used. All the statistics in Table III are provided using distributions represented as histograms.

There are six quantities for which we could not find suitable statistical information in the two sources mentioned earlier. These quantities are: (1) the probability that household members are living apart; (2) the probability that taxpayers are assisted by their spouses for realizing an income; (3) the distribution of taxpayers over countries of residence (noting that Luxembourg has a significant population of non-resident taxpayers); (4) the distribution of gross annual incomes for taxpayers (as opposed to those for households given by Row 6 of Table III); (5) the detailed breakdown of the > €1 million gross annual income bracket; and (6) the distribution of child allowances over taxpayers within individual households.

We addressed these gaps in our statistical information using feedback from legal experts. Specifically, for quantities (1) through (4), we defined heuristics for value assignment. For example, for quantity (3), we set the country of residence to Luxembourg for resident taxpayers; for non-resident taxpayers, we randomly picked one of the neighboring countries (France, Germany, or Belgium) as the country of residence.

With regard to quantity (5) and to avoid generating unrealistically large incomes, we put an upper bound on incomes larger than €1 million. Dealing with quantity (6) was slightly more involved. This is because taxpayers have some degree of control over who receives a child allowance within a household. This can affect the simulation results (see *is_receiving_ allowance_ for_children* input in Fig. 2). We tried to make quantity (6) as realistic as possible through optimization. In particular, we distributed child allowances over the taxpayers within the same household in a way that would minimize the household's overall tax obligation.

In total, we attached 57 annotations (stereotypes) to our domain model for guiding the data generation process. About 70% of the annotations came from public sources. The remaining 30% were based on feedback from experts and common sense, e.g., the optimization for quantity (6) above. In summary, we could successfully compensate for the lack of access to real data in our case study. Whether the data generated based on our annotations leads to credible simulation results is discussed in RQ2.

B. RQ2: Are the simulation results credible?

We used the statistics discussed in RQ1 to guide our data generator. We ran the data generator 10 times, creating 10 samples, each containing 10,000 taxpayers. The execution time for a single run of the data generator was ≈30 minutes. Once the samples were generated, we used the sanity checking procedure discussed in Section III-C to ensure that the samples were aligned with the known distributions for the statistics of Table III. Specifically, for each statistic in this table, we calculated 10 Euclidean distances, one per generated sample, measuring how close that particular sample was to (the distribution of) the statistic. In Fig. 5, we provide box plots for the Euclidean distances calculated for selected statistics from Table III. As shown by the figure, the maximum values in the box plots are < 0.1, suggesting that all the generated samples were aligned with the actual simulation population. Similar results were observed for the other statistics of Table III not covered in Fig. 5.

After ensuring the quality of the generated data samples, we ran our simulator (automatically derived from our policy models) over the samples. The simulation took approximately 80 minutes to process each sample. As noted in earlier sections, we answer RQ2 by simulating the status quo and measuring how aligned the simulation results are with the current tax contribution statistics. The current statistics pertain to the most

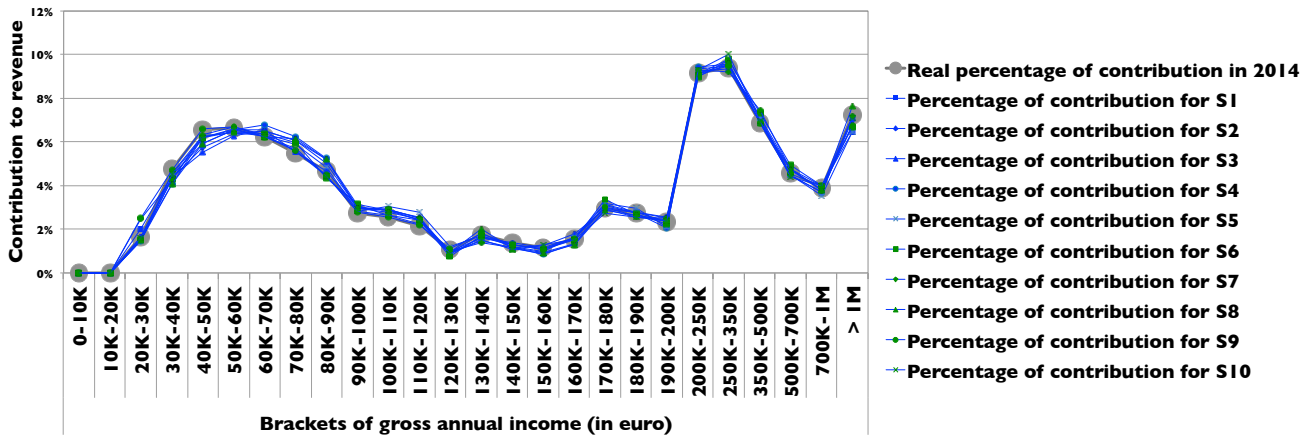

Fig. 4. Simulated and Actual Contributions of Households to Revenue

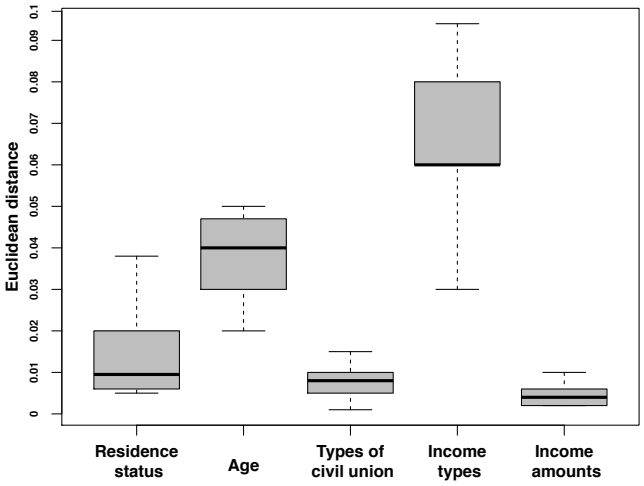

Fig. 5. Euclidean Distances between (Selected) Distributions of the Actual Simulation Population and Distributions of the Generated Data Samples

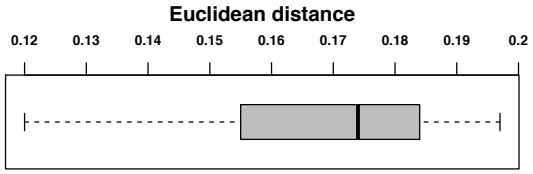

Fig. 6. Euclidean Distance Box Plot for Simulation Results

recent tax year that has been assessed (2014) [16]. In more precise terms, we address RQ2 based on the contributions of households in different income brackets to the overall income tax revenue collected by the Government. In the simulation report that we prepared for our public service partners [10], we additionally considered the distribution of taxpayers over tax classes, the distribution of households by tax brackets, the evolution of income taxes by households, and the evolution of the overall revenue. Since we do not have any statistics for these dimensions to compare our simulation results against, we do not use them in RQ2.

Fig. 4 shows the results from the 10 runs of our simulator alongside the current statistics from 2014. The x-axis of the chart in Fig. 4 represents the income brackets; the y-axis represents the contributions of households to the revenue. For example, the chart indicates that, in 2014, households having a gross annual income between €50K and €60k contributed $\approx$7% of the total tax revenue.

We observe from the chart of Fig. 4 that the simulation results are closely in line with the current statistics. To quantify how accurate our results are, we computed for each simulation run the Euclidean distance between the obtained curve and the current statistics (represented by the curve marked with large circles in Fig. 4). In Fig. 6, we present a box plot for the Euclidean distances calculated over the simulation results. The figure shows that our simulation results deviate from the current statistics by a distance ranging from 0.12 to 0.197 with a median of 0.174. We believe that these results, given the restrictions we had in terms of access to real data, are very promising. The positive results from RQ2 provide confidence that we modeled the legal policies at the right level of abstraction for our analysis, and further that our data generator, while not a substitute for real data, can still produce meaningful data for simulation.

C. Threats to Validity

The most relevant aspects of validity for our case study are internal and external validity, as we discuss below.

Internal validity. The statistical distributions that we used for data generation pertain to different taxation years. This raises the potential problem that some of the distributions may no longer be valid for the taxation year addressed in our case study (2014). To mitigate this problem, we used statistics only from the last ten years, our assumption being that the properties of the taxpayers population would not change drastically over a short span of time. The close alignment seen in the chart of Fig. 4 between our simulation results and the actual contributions of households to the revenue for the addressed taxation year suggests that using old distributions (from the past ten years) has not had a major negative impact on accuracy.

External validity. To date, we have applied our simulation approach in only one legal domain, i.e., taxation. Further case studies in other legal domains are thus essential for improving external validity. Our modeling notation is geared towards laws

that provide step-by-step instructions for performing legal administrative processes and achieving compliance. This makes us optimistic that our approach will generalize with ease to laws other than taxation that have the above characteristic, e.g., social security laws. Whether our approach can be extended to a broader spectrum of laws, e.g., those having a more declarative nature and relying heavily on deontic modalities such as permissions and obligations, needs a thorough investigation.

V. Lessons Learned

In this section, we reflect on the lessons that we learned from the case study described in Sections III and IV.

Align the models with the reasoning of legal experts. For a given policy, one can often come up with alternative but semantically-equivalent model representations. To illustrate, consider the policy model of Fig. 2. Here, one can merge the decisions concerning the age (*taxpayer_age* $\geq$ *64*) and the allowances (*is_receiving_ allowance_ for_children*) using a logical disjunction. Alternatively, one may swap the order in which these two decisions occur. In either case, we obtain a model that is semantically-equivalent to that in Fig. 2. An observation that we made during policy model validation is that legal experts usually have a preconceived mental image of the workflow of a policy. When a policy model deviated from this mental image, the experts proposed modifications to avoid logical formulas that were unnatural to them and to bring the workflow of the models in line with the way they reasoned about the policies. To improve communication with legal experts and increase understandability, it is therefore important not to view a policy model merely as a logical formula. Special attention needs to be paid to how the model is represented and ensuring that the representation is aligned with how legal experts think.

Maintain traceability to legal texts. Another observation from model validation is that, during the validation process, the experts frequently needed to consult the legal provisions underlying the policies. Being able to do so effectively requires traceability between the policy models and the relevant legal texts. In particular, the policy models and their inputs need to be traced to the specific legal provisions pertaining to them, e.g., articles in legislative texts. For example, the policy model of Fig. 2 is linked to articles 119 and 157bis of *LITL*, as shown on the top of the model.

A further remark about traceability concerns the relationship between the changes made to legal texts and the updates made to the policy models in response to these changes. (Proposed) changes to legal texts are typically provided as amendments. During our validation, we observed that legal experts could not readily follow how a change stated in an amendment related to the updates made to the policy models in order to materialize that change. This prompted us to develop the simple guidelines shown in Table IV to clarify how different types of amendments can impact the policy models. Subsequently, when we modified a policy model in response to an amendment, we explained to the legal experts the *rationale* behind the modifications by tracing the amendment to the affected model elements (including deleted ones). We received positive feedback from the experts about our clarifications and how we traced the amendments to model changes.

TABLE IV
Impact of Amendments on Policy Models

Amendment type	Description	Effects on policy models
Addition	New provisions, for example, new articles or paragraphs, are being added to the law.	Based on the nature of the new provisions, either new policy models are created or the existing ones are extended.
Deletion	Some existing provisions are being repealed and removed from the law.	Some existing policy models may be removed in their entirety or in part. No new policy specification is normally needed in response to a deletion amendment.
Replacement	Some phrases in existing provisions are being replaced.	The effect is that of combining additions and deletions described above. The effect of replacements is often limited to making changes in the policy models.
Redesignation	The numbers or titles of some existing provision are being changed.	No changes are required to the inputs or the workflows of the existing policy models. The traceability information that links the models to the text of law needs to be updated to account for the new numbers or titles.

Keep the modeling notation simple and lean. Once provided with adequate training, legal experts were able to grasp with relative ease class and activity diagrams. In contrast, OCL, which as noted in Section II-A is used for expressing the inputs to our policy models, posed a challenge to the legal experts, despite (basic) training and our attempts to explain the meaning of the OCL expressions. While we did not try to use any other formal language as a replacement for OCL and examine whether understandability would improve, we anticipate that using formal languages in general causes a communication barrier in the context of legal requirements. To validate our policy models with legal experts, we removed from these models all OCL expressions. To compensate, we wrote precise natural-language descriptions for the model inputs and used these descriptions as the basis for validation. The policy model of Fig. 2 alongside the glossary of Table I provide an actual example of the artifacts we used during validation.

A further observation is that although the legal experts in our case study could understand and review class and activity diagrams, they could not be expected to keep the notational details in their working memory for long. In the early validation rounds, we would start a session with a brief "refresher" on the modeling notation, e.g., reminding the experts that the diamond shapes represent decisions, the *«update»* actions store simulation results, and so on. In later rounds, we prepared a *notation legend* and added it at the bottom of the models being validated. We found this strategy to be very effective for facilitating communication with legal experts. An important consideration with regard to legends is for them to be as succinct as possible, implying that there is an advantage in restricting the notational elements being used to what is absolutely essential for expressing the models.

VI. Related Work

Model-based techniques are increasingly being used for expressing legal requirements. For example, van Engers et al. [17] use UML for modeling the Dutch tax legislation. Ghanavati et al. [1], [18] employ a combination of goal models and use cases for capturing legal requirements and supporting the comparison of multiple regulations. Ingolfo et al. [2], [19] develop a goal-oriented modeling framework for arguing about regulatory compliance. Breaux and Powers [20] specify legal requirements via business process models and use these models for compliance checking. Zeni et al. [3] and Breaux [4] develop conceptual models for characterizing key abstractions in legal texts, and exploit these models for ambiguity reduction and various types of automation. These earlier work strands are not specifically targeted at simulation. Our work complements the above strands by proposing an approach for modeling legal requirements as executable policies and simulating these policies.

Several tools exist in the field of applied economics for policy simulation, e.g., SYSIFF [21], POLIMOD [22], and EUROMOD [23]. These tools use a combination of spreadsheets and software code for specifying legal policies. This strategy can complicate the validation of the resulting policy specifications, as legal experts often lack the software engineering expertise required to understand complex spreadsheets and software code. In contrast, our approach uses models to raise the level of abstraction at which legal policies are specified. This helps improve the understandability of policy specifications by legal experts. Furthermore, the above tools assume that the input data for simulation is available a priori. This assumption does not always hold, as we noted in Section I. Our approach is equipped with a built-in data generator that can produce artificial but representative input data for simulation when real data is missing or incomplete.

VII. Conclusion

In this experience paper, we reported on a real-world case study of our model-based approach for simulating legal policies. The technical foundations of the approach were developed in our recent previous work [9]. Through our case study, we evaluated the feasibility and usefulness of our approach in practice to assess changes to policies, demonstrating that the approach can be applied with reasonable effort, and that it yields credible results. We further discussed the lessons we learned from the case study, particularly in relation to making modeling more palatable to legal experts. An important characteristic of our simulation approach is that it builds on the same models that are used for legal compliance analysis. This makes the experience gained from our case study relevant not only to simulation but also to legal compliance.

In the future, we plan to use our simulation approach to conduct case studies beyond the taxation domain. For example, we would like to apply the approach to social security and customs laws where simulation is commonly used for assessing the risks and consequences of legal reforms. We would further like to extend our approach to account for situations where the simulation input data needs to be dynamically updated, e.g., when taxpayers change their behavior in response to changes in the laws and regulations.

Acknowledgment. We thank members of Luxembourg's Inland Revenue Office and National Centre for Information Technologies (CTIE), particularly T. Prommenschenkel and L. Balmer for sharing their valuable time and insights with us. Financial support was provided by CTIE and FNR under grants FNR/P10/03 and FNR9242479.

References

[1] S. Ghanavati, A. Rifaut, E. Dubois, and D. Amyot, "Goal-oriented compliance with multiple regulations," in *RE'14*, 2014.

[2] S. Ingolfo, A. Siena, and J. Mylopoulos, "Nòmos 3: Reasoning about regulatory compliance of requirements," in *RE'14*, 2014.

[3] N. Zeni, N. Kiyavitskaya, L. Mich, J. R. Cordy, and J. Mylopoulos, "GaiusT: supporting the extraction of rights and obligations for regulatory compliance," *REJ*, vol. 20, no. 1, 2015.

[4] T. Breaux, "Exercising due diligence in legal requirements acquisition: A tool-supported, frame-based approach," in *RE'09*, 2009.

[5] W. Emmerich, A. Finkelstein, C. Montangero, S. Antonelli, S. Armitage, and R. Stevens, "Managing standards compliance," *IEEE TSE*, vol. 25, no. 6, 1999.

[6] T. Breaux, "Legal requirements acquisition for the specification of legally compliant information systems," Ph.D. dissertation, North Carolina State University, 2009.

[7] N. Sannier, M. Adedjouma, M. Sabetzadeh, and L. C. Briand, "An automated framework for detection and resolution of cross references in legal texts," *REJ*, 2015.

[8] G. Soltana, E. Fourneret, M. Adedjouma, M. Sabetzadeh, and L. C. Briand, "Using UML for modeling procedural legal rules: Approach and a study of Luxembourg's Tax Law," in *MODELS'14*, 2014.

[9] G. Soltana, N. Sannier, M. Sabetzadeh, and L. C. Briand, "A model-based framework for probabilistic simulation of legal policies," in *MODELS'15*, 2015.

[10] L. C. Briand et al., "Policy simulation in action: A case study on how the potential abolishment of joint taxation will impact personal income taxes," SnT, University of Luxembourg, Tech. Rep. TR-SnT-2016-2, 2016, http://people.svv.lu/soltana/Simulation_Report_2016.pdf.

[11] Object Management Group, "Object Constraint Language v2.4 (OCL)," http://www.omg.org/spec/OCL/, 2014, last accessed: February 2016.

[12] C. Larman, *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition)*. Prentice Hall, 2004.

[13] Object Management Group, "UML Superstructure Specification," 2009.

[14] G. Soltana et al., "Using UML for modeling legal rules: Supplementary material," SnT, University of Luxembourg, Tech. Rep. TR-SnT-2014-3, 2014, http://people.svv.lu/soltana/Models14.pdf.

[15] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *Mathematical Models and Methods in Applied Sciences*, vol. 1, 2007.

[16] Luxembourg's Ministry of Finance, "Compendium sur les données statistiques des impôt luxembourgeois," 2015.

[17] T. van Engers, R. Gerrits, M. Boekenoogen, E. Glassée, and P. Kordelaar, "POWER: using UML/OCL for modeling legislation - an application report," in *ICAIL'01*, 2001.

[18] S. Ghanavati, D. Amyot, and L. Peyton, "Towards a framework for tracking legal compliance in healthcare," in *CAiSE'07*, 2007.

[19] S. Ingolfo, A. Siena, J. Mylopoulos, A. Susi, and A. Perini, "Arguing regulatory compliance of software requirements," *Data & Knowledge Engineering*, vol. 87, 2013.

[20] T. D. Breaux and C. Powers, "Early studies in acquiring evidentiary, reusable business process models from laws for legal compliance," in *ITNG'09*, 2009.

[21] L. Canova et al., "SYSIFF 2006: A microsimulation model for the French tax system," Paris School of Economics, Tech. Rep., 2009.

[22] H. Sutherland, "The development of tax-benefit models: a view from the UK," University of Cambridge, Tech. Rep., 1995.

[23] F. Figari, A. Paulus, and H. Sutherland, "Microsimulation and policy analysis," *Handbook of Income Distribution*, vol. 2, 2014.