Digital Playground for Policy Decision Making

© J. Hedtrich

© C. Henning

© E. Fabritz © B. Thalheim

Kiel University

Department of Agro-Economy and Department of Computer Science,

Kiel, Germany

johannes.hedtrich@ae.uni-kiel.de chenning@ae.uni-kiel.de ef@is.informatik.uni-kiel.de thalheim@is.informatik.uni-kiel.de

Abstract. Policy development is a complex and highly dimensional process. This complexity is very difficult to comprehend due to complexity of the parameter space, multi-dependence of parameters, and the nature of process. Therefore, policy makers should be supported while considering and evaluating various alternative decisions. This paper illustrates a modelling approach for advisory and assistance in decision making for political practitioners. We describe the corresponding advisory tool supporting the interactive decision process.

Keywords: Computer-based communication tool, interactive learning between scientific models and practitioners, political decision making support

1 Introduction

Policy decision making is a complex task which comprises the understanding of possible positive or negative consequences of decisions as well as a mechanism to restore consistency of a system in the case of inappropriate decisions. Thus, even policy experts often have only a vague understanding of how policies impact on relevant outcomes. Therefore, political practitioners use simple mental models (beliefs) to understand complex impacts of policies. For this reason, a technical solution for the simulation of policy impacts can be helpful, e. g. a graph displaying the impact of parameters. Our software will work as a digital playground system with relevant decision parameters as inputs and implied outcomes (consequences of the decision) as outputs.

Nowadays it is commonly accepted that good economic policy has to be evidence-based, i.e. rest on scientific knowledge and statistically proven evidence. However, scientific modelling is often criticized by political practitioners as a purely academic exercise that fails to provide practical tools for understanding or designing optimal real-life economic processes [5]. Accordingly, scholars promote participatory policy analysis that is characterized by an interaction between economic theory and political practice to combine the 'objective' knowledge derived from economic theories and empirical data with the 'subjective' knowledge of stakeholder organizations as political practitioners ([2], [9], [5]). Moreover, inadequate communication between scientific policy analysts and political actors is proposed to be a principal cause of the limited impact of research

on policymaking. For example, the 'utilization of knowledge school' emphasizes the fact that policy analysts and policymakers live in two separate communities [5]. Hence, to become more efficient, the relationship between scientific experts and policy actors must be redefined.

Moreover, Stiglitz argues in his highly recognized book "Whither socialism?" [14] that the market-socialist experiences in Eastern Europe failed due to the incorrect beliefs of politicians in the Arrow-Debreu concept of real market economies as a complete set of competitive markets ([14], Chapter 11). Interestingly, Stiglitz's explanation of the failure of the market socialism experiment highlights an interesting general point: economics must be recast as something more than a constrained maximization problem to understand and design real economies. In other words, theoretical models provide a relevant benchmark for understanding real-life economic processes but require abstract scientific models and political praxis to actually change the world. Hence, as previously discussed in [6], [12], [7], identifying effective solutions for central economic problems appears to be a problem of linking abstract economic theory with feasible political practice. Accordingly, scholars of participatory policy analysis discussed innovative tools, such as participative modelling [5] (i.e. improving communication in formal models by means of interactive or man-machine simulations [1] or decision seminars [10]).

Beyond interesting methodological ideas and concepts for assessing the role of relevant 'objective' scientific knowledge it is important to better understand and design the complex communication processes between science and political practitioners in a way that combines the knowledge of both worlds

174

Proceedings of the XX International Conference "Data Analytics and Management in Data Intensive Domains" (DAMDID/RCDL'2018), Moscow, Russia, October 9-12, 2018

to generate advanced solutions to existing economic problems, such as the transformation to a sustainable bio-economy or reaching sustainable development goals.

In this context the paper develops a computer-based tool Policy-Lab that facilitates an interactive communication and learning between political practitioners and scientific models. Figure 1 shows a graphical representation of positioning of scientific models statements (scientific world), political practitioners beliefs (stakeholder beliefs world) and the aspired communication between these two worlds.





2 Policy-Lab tool

The Policy-Lab tool has to fulfill various tasks in order to effectively support the decision-making process and facilitate the learning of stakeholders. Those tasks can be characterized with following devices:

- Input Device: Survey policy preferences, goals and beliefs using questionnaires.
- Report Device: Report surveyed data back to the group. This requires the dynamic application of statistical analysis of the data.
- Interactive Modelling Device: Users can simulate different policies and evaluate their impact on policy goals.
- Consensus Device: This device provides support in finding a potential political compromise.

Political practitioners in the area of economic policies have to choose a policy that best achieves their desired policy goals. This relationship is usually highly complex and involves making trade-offs between goals. Therefore, models are needed to better understand this relationship. The application of such a model is an integral part of the previously described devices. An example of a model capturing this complex relationship is shown in the following:

$$\gamma = [\gamma_i]; i \in I = \{1, \ldots, n_i\}; s \in S = \{1, \ldots, n_s\};$$

$$j \in J = \{1, ..., n_j\}; \ t \in T = \{0, ..., 9\}$$
$$Be_s(\gamma) = \eta_s \left(\sum_{i \in I} \mu_{i,s} \gamma_i^{-\rho}\right)^{-\frac{1}{\rho}}$$
$$tp_s(Be_s) = tp_s^{max} \frac{e^{a_s Be_s + b_s}}{e^{a_s Be_s + b_{s+1}}}$$
$$wZ_j(tp_s) = \xi_j^{const} + \sum_{s \in S} tp_s \cdot \xi_{s,j}$$
$$Z_{j,t}(wZ_j) = Z_{j,0} \cdot (1 + t \cdot wZ_j)$$

The formulas describe the transformation of a chosen policy, i.e. a budget allocation γ to policy programmes, into policy outcomes over time $Z_{j,t}$. Poverty reduction is an example for a policy goal. In the first step a CES (constant elasticity substitution) function is used to transform allocated budget into sector specific effective budget Be_s . The effective budget is then transformed into a change in technical progress tp_s using a logistic function. The change in technical progress is then transformed into growth rates of goals wZ_j , and in the final step the goal achievement levels $Z_{j,t}$ are computed.

In this case developing a decision based on mathematical formulas is rather difficult for some stakeholders, that is why some supporting technical solutions are necessary.

To support the simulation of these models technical methods and frameworks are used. The methods being used during the simulation and estimation are mathematical statistical methods (Bayesian model averaging, Meta Modelling) and programming languages for statistical computing (R) and optimization problems (GAMS). In our case a meta model is a surrogate for a more complex model, which captures the main relations in an explicit mathematical form.

In order to make models accessible to a wide range of users, who most commonly are organizations or individuals, interested in the construction of economic policies (in our case these are agricultural policies [8]), an intuitive visualization is required. The visualization part of the tool should work as a playground for model simulation supporting expert learning, model learning, interactive learning (expert-model-expert exchange), and learning from collective decision (voting over policies or exchange games).

Thus, the Policy-Lab tool should work as an interactive input-output playground for the models simulation and graphical visualization.

At the same time, the tool should process a large amount of model specific data: different kinds of inputoutput parameters and computational cores of models. So an important issue during the tool development is the implementation of a suitable database structure.

The Policy-Lab tool will be implemented in the form of a web application. The tool is now in the creation phase, for this reason the main concepts of tool development, tool requirements, and its structure will be discussed further.

2.1 Theoretical concepts

To begin with, some theoretical concepts will be explained:

1) What is a model from the tool's perspective?

In the sense of the current tool, a model is a computable unit with defined input parameters, computational core, and computed output parameters, which can be shown in a graphical form. A special sub-type of a model is a questionnaire, that has input parameters and computational core, which adds user input to a statistical model and recalculates its output. The output of recalculation is not shown to the users directly, but should be available in another view.

2) How model data will look like?

The computational core of a model is predefined by the model scientists. It can be written in R, GAMS or in other programming languages. The input and output parameters depending on the language used are language specific character values, which can be saved in a database or in an external file. This data should be accessible to the playground.

2.2 Playground system requirements

The creation of the simulation playground begins with the comprehension of its required features. Partly this information can be derived from the existing Policy-Lab tool prototype, partly from model scientists' requirements and user expectations.

The list of requirements for the simulation playground includes the following:

- clear and comprehendible software structure
- clear and comprehendible database structure
- scalability of the system
- maintainability of the system
- efficiency of the system
- run-time reciprocative input-output system
- user-friendliness of the system

Based on the analysis of system requirements the following issues can be defined during the development of the tool:

- How to implement interactive forms for userinput and output? Which interfaces are needed?
- How input-output parameters and computational cores of models look like and how they are saved?

• How the communication between the computational core of a model and the web interface looks like?

2.3 Playground system structure

The playground tool should serve as a web information system for model simulations with interactive inputoutput mechanisms for users. The system should have a clear structured database, expandable for new entities, since the system will describe a varying number of models. The system should visualize a list of models and its descriptions for users. Further, the system should have views for input parameters from users and possibilities for the graphical representation of computed output. Another integral part of the system is a computational module, where the computation of output takes place.

According to the system description and requirements the new system should have the following components:

- Web interface for users with possible use-cases definition, user management functions, presentation of views related to models, including model-input parameters and output graphics.
- Computational module with possible integration of R and GAMS sub-modules.
- Communicational interface: beside other functions web interface and computational module should be capable of interaction with each other.
- Database for the web interface
- Database for the computational module

Web interface

Web interface is a unit that contains common login, logout, and register functions, explanative use-cases, overview of present models, view for input parameters of the models, and view for the output in graphical form. Moreover, there should be a separate view for administrators to allow user management.

Computational module

Computational module is a unit that can be connected to R or GAMS sub-modules or use some other programming language for computation. This module should communicate with the web interface: parse userinput parameters, convert them to input parameters in the format of computational language depending on the model, and parse computed output back to the chosen web interface format.

Database for the web interface

Database for the web interface should contain all the information about users and their management, widgets shown in the interface, and shown model views. Furthermore, for the representation of input and output this database should have information about input and output parameters of a model.

Diagram 1 shows a fragment of a possible ER-schema of the database:



Diagram 1

The ER-schema describes users, their roles, and interfaces that depend on roles. Further, the schema includes descriptions of models, their simulations and different types of simulation result parameters. Additionally, every interface page has specific widgets of different types depending on model being simulated, including charts and questionnaires.

Database for the computational module

In the case computation is produced in another application a separate database is needed.

The database for the computation should have information about models, their computational cores, and their input-output parameters.

If the computational module does not need its own database, analogical database entities are necessary.

A possible ER-schema of a computational module is shown in Diagram 2:



Diagram 2

Communication between web interface and computational module

Communication between these two modules is an important part of the system, the whole software structure and efficiency depends on the form of communication.

Two architectural alternatives for modules communication have been developed:

1) Web interface and computational modules can be placed inside of one software project, so that the division in interface and computation is only a logical notion. In this case the interface and computational parameters can be saved in the same database. The computation itself can be made, e.g. with JavaScript language. In the case of JavaScript, the computation will proceed efficiently as no integration of external R and GAMS modules is needed. The communication in this case is trivial and proceeds within one application.

2) In the other case, R and GAMS modules can be stored in a separate application with an independent database. In such a case computation needs these modules because of computational complexity. Moreover, the separation of computational component allows to bring a modular structure to the software. In addition, the exchange of or changes in R or GAMS models are made easier, because they do not influence the execution of the web interface in a negative way. Thus, the two components are not only logically, but also physically separated from each other. Each component has its own database. The communication between the web-interface application and the application where model computation takes place proceeds with HTTP-messages, containing input-output parameters for computation and information about models in JSON format.

Figure 2 illustrates how this kind of architectural style can be implemented:



Figure 2

In the system the both ways of communication will be used, depending on the complexity of a model.

2.4 Advantages of the system

The described playground system has a number of advantages:

- The system is scalable and extendable, as the underlying web information system is dynamic and is built accordingly to the database contents. The expandable database allows the insertion of new visual elements and models for the simulation.

- The first architectural style for communication allows the implementation of a run-time reciprocative inputoutput system.

- The second architectural style for communication contributes to system's modularity and can be approached from two different perspectives: web interface based and computation based perspective. Thus, two scientists can work simultaneously on the two components. Any changes in one of the components would not cause error or stoppage of the execution in the other component. After the adaption of communicational modules, the changes can be accepted by both components.

- The tool supports expert, model, and interactive learning, moreover the learning from collective decision is implementable.

- Description of use-cases supports user-friendliness.

2.5 Practical example of playground usage

Economic models consist of mathematical formulas, which describe a large amount of economic factors and their correlation. Such models are developed by scientists and are often difficult to understand for policy makers. A policy maker would like to observe the influence of economic factors on the formulated model without analysing the complex correlation of these factors. For this reason, a suitable solution can be the graphical view of the model input parameters with the possibility to change them and observe the graphical output depending on these inputs. Thus, the program interface built in a similar way as described above helps to identify the tendencies in interrelations between input factors and output results as well as to formulate a forecast of the development. A prototype of such a kind of program interface was implemented in the playground.

A real-life example of model simulation inside the playground will be shown further.

In the first step, the simulated model will be described. In addition, the views of the implemented prototype of model simulation will be illustrated.

The simulated model is a simplified version of a larger model, which covers the relation between the investment in economic policies, with a special focus on agricultural policies under the CAADP (Comprehensive Africa Agriculture Development Programme⁷) framework, and different policy outcomes, like poverty reduction, income of farmers or urban consumers. Political practitioners are faced with the decision which policy to choose, i.e. how much money they should

spend and how to distribute it. They want to choose a policy that best achieves their policy goals. The described link between policies and goals is very complex and is not straightforward. Therefore, the simulated model and its implementation into the playground provides the first step in helping them better understand the results of choosing a specific policy and validating or updating their beliefs on this relation.

The model is derived using a Bayesian estimation procedure and it combines statistical data and expert data to estimate the parameters of the model. Some formulas for the simplified model along with a short technical explanation are shown at the beginning of section 2.

The input parameters are chosen in a nested, topdown approach. The first decision specifies how much of the total state budget should be invested into economic policies (invest). The second one defines how much of that money should be put into agricultural policy programs (agrar), with the remainder being invested into non-agricultural policy programs. The last decision is how the agricultural budget should be distributed between the four main investment pillars: Natural Resources (pi_nr), Farm Management (pi_fm), Market Access (pi_ma) and Human Resources (pi_hr). Each input parameter has a predefined range and step on which the parameter can be decreased or increased.



Figure 3

The changing of parameter values can be made by users with the help of slider-widgets (See Figure 3).

The computational core of the model is written in JavaScript which corresponds to the first architectural style (See 2.3). This computational core comprises functions to transform the previously described input into different indicators for the output.

The outputs of the functions from the computational core are represented with the chart-diagrams. These diagrams were built with the help of Highcharts library.

Different output sets of the model computation are accessible through correspondent tabs (here: gamma, policy, pillar_budget, effective_budget, tp, wz, z1 through z7). The output sets show on the one hand the partially transformed input: first into gamma, then into policy and then into pillar_budget, showing the relative and absolute budget shares. Then there are technical indicators like tp and wz, that show the additionally achieved technical progress and growth rates for the policy goals. And the z1 through z7 show the measures

⁷ http://www.un.org/en/africa/osaa/peace/caadp.shtml

for the different policy goals in real measures. For example, z2, poverty reduction, as the percentage of the population living under the poverty line. Each tab with correspondent output set can have different graph types (column or line) as output.



Figure 4



Figure 5

The first interface shown in Figure 4 displays the growth rates for the different policy goals. Another important feature can be seen here, is the comparison between two output graphs. The output calculated with the previous input values is shown with the grey colour in the central and the right windows. The output generated from the actual input values is illustrated with in the blue chart.

The second figure represents the development of the z3, the provision of public goods indicator, over time as a line-chart. The comparison between the two outputs is made analogue to the previous figure.

As it is shown in the pictures the slider-widgets are used to set the value of the input parameters. Thus, a user can regulate input parameters by moving sliders and see the effect of parameters changing in the chart. The charts contain the output diagram computed from the input values newly set and the output diagram (grey colour in the central window) computed from the old input values (or values from the previous slider-state). Moreover, the old diagram output is shown in a separate window beside the corresponding old input values (right window).

The output graph in the right window with corresponding input values can be fixed on the page, so that changing of input parameters will have no influence on the fixed output graph and graphs with newer values will not rewrite it.

This example shows an interactive solution for setting user-inputs and receiving graphical outputs. The

old chart values representing the previous state allow to visualize the output changing depending on different inputs. Thus, the development tendency of a model can be comprehended by the users.

3 Conclusion

Described Policy-Lab tool facilitates political decision making by presenting an interactive playground system, that simulates a large opportunity space for policy decisions and computes possible effects of the model simulation with the decisions made.

As a result, Policy-Lab tool for policy decision enables political practitioners to relate potential policy decisions to corresponding outcomes.

The described tool should be flexible, efficient and user-friendly, in order to be able to simulate the full complexity of the models and to assist in successful decision making.

Related work

There exist other systems, which work with interactive user input-output and use a large number of possible input parameters and calculations, beside the Policy-Lab tool prototype, the precursor of the current simulation tool, mentioned above.

Examples of agricultural frameworks:

FAPDA Web-based Tool [4] provides a decision making framework for food and agricultural policy decisions.

Another decision making GIS-based tool is ReSAKSS [13], it contains data on agricultural, socioeconomic and bio-physical areas. This tool assists policy makers in developing agricultural policies.

Examples of other frameworks:

Today one can find modelling tools which accept a wide range of parameters and simulate some complex process in order to understand the influence of these parameters on the system in medicine.

The Lives Saved Tool for Maternal and Child Health (LiST) [15], [11] is a modelling framework developed by the Institute for International Programs at Johns Hopkins Bloomberg School of Public Health with intention to estimate the effect of health coverage on maternal and child health. LiST models the status of health coverage under the influence of various factors (e.g. increasing of health care services and usage of nutrition interventions). In this tool users can estimate the impact of different kinds of heath interventions in order to plan the strategies for the improvement of medical methods supporting maternal, newborn, and child health. The tool contains the data about the effect of some kinds of interventions on people's health. Further, the data about maternal and newborn mortality rates, health coverage and interventions of a particular country or

region is collected. Thus, a user can simulate the usage of specific health care methods in a particular region and see the influence of this usage as graphical output.

The Multi-Criteria Analysis Decision framework is a modelling framework for decision making and priority setting, which elaborates on possibilities to create "an equitable, efficient, and sustainable health care system" [15]. All possible health interventions are ranked and compared during a multi-criterion analysis. A specific web-based framework to implement this approach was developed by the EVIDEM Collaboration [3]. The EVIDEM tool is used to provide the participants of the health care process with information and to support decision making during this process. The tool simulates different factors influencing patients' health and produces a graphical output measuring the importance of these factors or the degree of their positive or negative impact.

References

- Duke R. D., Origin and Evolution of Policy Simulation: A Personal Journey. Simulation & Gaming. XX(X) I – 17. SAGE Publications (2011)
- [2] Durning D. Participatory Policy Analysis in a Social Service Agency: A Case Study Journal of Policy Analysis and Management, JSTOR, 12, 297 (1993)
- [3] EVIDEM.
- [4] https://www.evidem.org/wp/wpcontent/uploads/2017/09/EVIDEM-10th-Edition-Tutorial.pdf
- [5] FAPDA Web-based Tool.
- [6] http://www.fao.org/in-action/fapda/web-basedtool/en/
- [7] Geurts J. L.A., Joldersma C. Methodology for participatory policy analysis. Department of Policy and Organization Science, Tilburg University, P.O. Box 90153, 5000 LE Tilburg, Netherland. European Journal of Operational Research 128 300-310 (2001)

- [8] Habermas, J. Theory and Practice. Beacon Hall, Boston (1974)
- [9] Hayek, F. A. The Counter-Revolution of Science. Liberty Classics, Indianapolis (1979)
- [10] Henning C., Badiane O., Krampe E. Development Policies and Policy Processes in Africa. Springer International Publishing (2018). DOI: 10.1007/978-3-319-60714-6. ISBN 978-3-319-60713-9. ISBN 978-3-319-60714-6 (eBook).
- [11] Joldersma, C. Participatory Policy Making: Balancing between Divergence and Convergence
 European Journal of Work and Organizational Psychology, Informa UK Limited, 6, 207-218 (1997)
- [12] Lasswell H.D. The structure and function of communication in society. <u>http://www.irfanerdogan.com/dergiweb2008/2</u> <u>4/12.pdf</u>
- [13] Lives Saved Tool. http://www.livessavedtool.org//how-listworks#Interventions
- [14] Offe, C. The Divergent Rationalities of Administrative Action. In Disorganized Capitalism, C. Offe. MIT Press, Cambridge (1985)
- [15] ReSAKSS.
- [16] http://eatlas.resakss.org/
- [17] Stiglitz J. E. Whither Socialism? MIT Press, (1996) ISBN 0262691825, 9780262691826, 338
- [18] The National Academics of Sciences Engineering Medicine. Country-Level Decision Making for Control of Chronic Diseases: Workshop Summary (2012).

https://www.nap.edu/read/13337/chapter/5