A step by step guide for developing agent-based SimPaSI models

Alex Smajgl

Analysing Pathways to Sustainability in Indonesia

www.csiro.au

BAPPENAS

Enquiries should be addressed to:

Dr Alex Smajgl
CSIRO, University Drive, Townsville 4810 QLD, Australia
p: +61 (0)7 47538615   m: +61 (0)419 793439

Copyright and Disclaimer

Important Disclaimer

Contents

Glossary

Bappenas	
Bappeda	
Bottom-up modelling	This describes an approach of modelling that develops the model from the level of disaggregated units (i.e. individuals or households). In contrary, top-down approaches develop the system representation from a highly aggregated level (i.e. sectoral production in CGE or IO modelling).
Ex-ante	Defines a period *before* a reference date (such as current year).
Ex-post	Defines a period *after* a reference date (such as current year).
GUI	Graphical User Interface
Pending decisions	Decisions that are discussed and likely to be made in the near future.
Participatory modelling	Defines a process of model development that involves stakeholders from the very first step of design.
SimPaSI	Simulating Pathways to Sustainability in Indonesia
UML	Unified Modelling Language, which was developed as a standard design for software development purposes in object oriented programming environments. It defines classes (entities), their main variables (attributes), and linkages between classes.

Acknowledgements

The author is grateful for the constructive comments provided by Nadine Marshall. The author wish to thank AusAID, CSIRO, and PROFOR/World Bank for funding this research.

1. INTRODUCTION

The aim of this document is to provide a manual for the development of empirical agent-based models. While many steps describe generic tasks the context this document is developed for is the specific situation of Indonesia and the continuation of agent-based policy analysis focused on developing further implementations of the SimPaSI (Simulating Pathways to Sustainability in Indonesia) model.

Agent-based models (ABMs) are computational models which contain an explicit and individual representation of the entities of the target system being modelled and of their interactions (Gilbert 2008). Agents in the model can represent individual entities such as humans with various levels of cognitive capacity, as well as groups of individuals and non-cognitive environmental entities (i.e. water, trees). As the system representation is developed from the perspective of individual entities (bottom-up approach) agent-based modelling allows for the analysis of "evolving systems of autonomous interacting agents" (Tesfatsion 2002). As Deadman (1999) points out, instead of defining the overall behaviour, in ABMs "overall behaviour emerges as a result of the actions and interactions of the individual agents." This makes agent-based modelling effective in analysing complex adaptive systems (Miller and Page 2008). In-depth descriptions of agent-based modelling can be found in Gilbert (2008).

The SimPaSI modelling approach assumes a participatory design (Smajgl and Prananingtyas 2009), in which the design phase is directed by relevant stakeholders. Additionally, it assumes that multiple decision-making-levels determine the outcome of a relevant problem, such as poverty or environmental degradation (Smajgl 2009). Thus, participatory activities are conducted with multiple decision making levels, such as central government, provincial government and district government representatives. Depending on the context other stakeholders need to be involved, such as businesses or NGOs.

Of critical importance is the understanding that models developed according to the manual do not aim for precise predictions. Instead, agent-based modelling is perceived as a tool that effectively facilitates discussions between diverse decision-making-agencies. Such a process aims for testing beliefs decision makers hold regarding potential impacts of available policy options on indicators relevant to them. While socio-ecological systems mostly fall in the domain of complex systems (Miller and Page 2008) human cognitions reduce often existing complexity to a degree that shapes unfounded expectations. Such expectations are challenged in workshop situations or meetings by confronting decision makers' beliefs with each other and with simulation runs and their results.

This document concentrates on the technical side of the modelling process without explicitly discussing the participatory activities. Figure 1 visualises the principle steps of model development, which also defines the structure of this document. Dashed lines symbolise likely loops while continuous lines define the principle sequence of steps. The participatory process (and the dotted line) require context-specific steps.

In a first step the case study specific problems have to be understood. Conducting workshops with district and provincial government officials to ensure their participation is essential. This process has to lead to an agreed list of policy options the model will be able to assess and a list of indicators involved decisions makers use to judge if outcomes are successful or not.

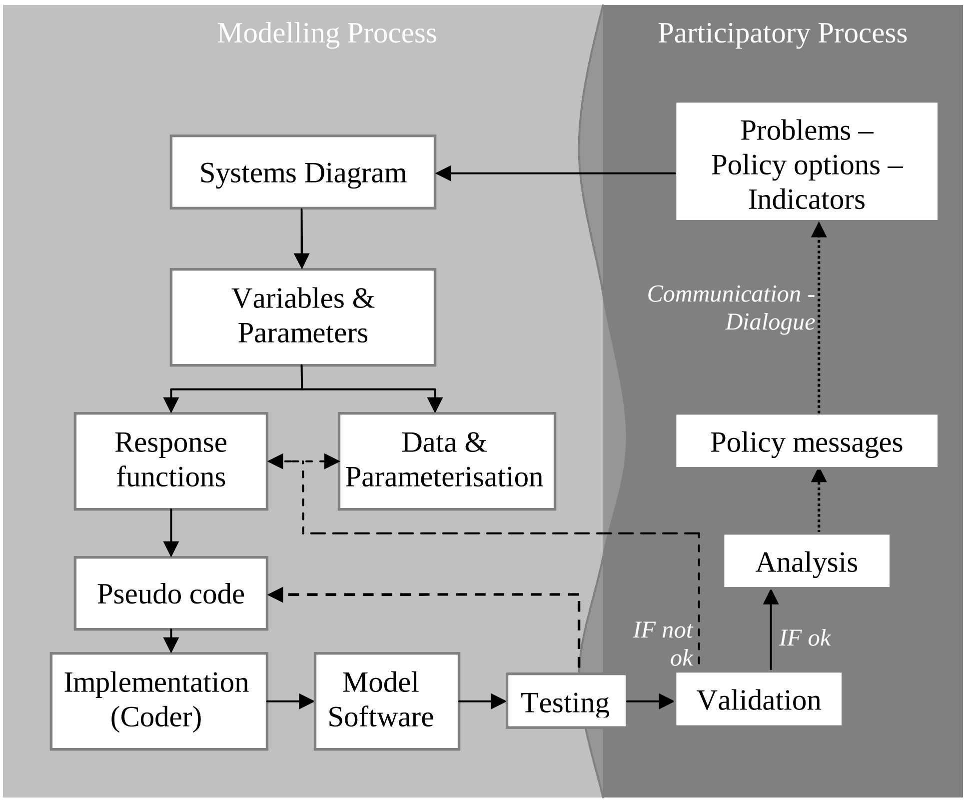


Figure 1: Flow-diagram of model development process

Once policy options and indicators are agreed a system diagram has to be developed. For each system element the modeller needs to decide if its state needs to change endogenously, which makes it a variable, or exogenously, which makes it a parameter. Together with experts response functions have to be developed for all variables and data has to be elicited for parameters (and for initial states of variables). Then pseudo code needs to be developed and handed over (in form of a design document) to a coder for implementation. The software needs to be tested and validated before actual analysis can be conducted and lead translated into policy messages. The following explains the modelling process in more detail.

2. PARTICIPATORY REQUIREMENTS

The first step in preparation of model design is the development of a functioning dialogue and partnership with the local decision makers. The process development should be focused on those local decision makers that are most relevant to the indicators considered by Bappenas and most relevant to the future trajectory of the region's development. The following elements define main questions that can guide the design of this process but do not a define comprehensive list (which depends on the context of each case study):

- **What are the relevant pending decisions for the central government?**

This element requires a discussion with central government decision makers to identify the policy decisions the model needs to assess. This information will later provide the basis for scenario definitions. The more precise the policy options can be defined the better the design can proceed. For instance, ‘fuel subsidy change’ is not sufficiently specific. ‘Increase of petrol prices by 19% on 1 July 2010 compared to current price levels’ is the required level of precision.

- **What are the relevant sustainability indicators for Bappenas?**

Indicators are relevant if used by involved decision makers to judge if a decision is a success or not. Sustainability means that the modeller should aim to identify indicators that define long-term success, if possible, across the triple-bottom line.

This step requires a discussion within Bappenas to develop a full understanding of in-house needs. A typical indicator is ‘poverty’ defined as the number of people below the poverty line. Additional long-term environmental indicators that allow projecting impacts on specific livelihoods and hence impacts on poverty are forest cover, fish population or the stock of another natural resource. Additional long-term social variables can be migration, education or other elements that impact on poverty changes. It is absolutely crucial to develop an exact definition of these indicators as otherwise data elicitation and model development are likely to provide the wrong type of information.

- **What are the relevant sustainability indicators for Bappeda?**

Having a principle understanding of policy scenarios and indicators relevant to the central government is crucial. Additionally, it is fundamental to capture the pending or potential decisions of local authorities. Local governments develop and invest in strategies to achieve specific development goals. Such decisions can contradict or enhance the impact of central decisions on relevant indicators. To avoid the misinformation of central decision makers it is important to capture the combined effect of central and local decisions. Capturing such multiple levels of decisions is one of the strengths of agent-based models. Therefore, a dialogue needs to be opened with Bappeda Provinsi and Bappeda Kabupaten and/or Bupati to bring together all relevant policy options. Additionally, the process should deliver all indicators that are relevant to local decision makers. Otherwise, no relevant information can be fed back into the region and the process is unlikely to gain any support from the case study region.

- **Who are the local experts?**

Two reasons emphasise the importance for involving local experts. Firstly, some local experts have good links to local decision makers by providing an advisory role. This means that the participatory process needs to understand such experts as door keepers. Secondly, the development of a system diagram and the elicitation of data through field work requires local expertise. Identifying local experts, such as university staff, with experience in the problem domain and experience in conducting field work will help create effective conditions for model development.

At the end of this process a robust list of specific policy options and indicators will be completed. As long as this list has not been confirmed this iterative process should be continued before starting the next step.

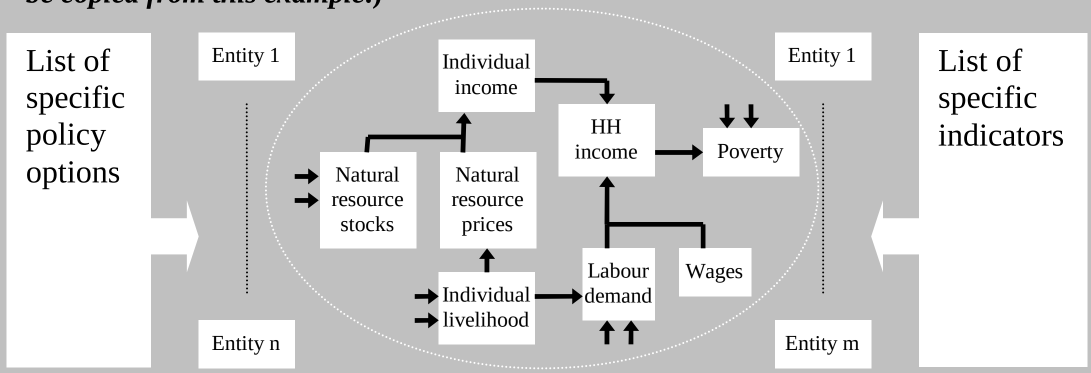
Example for output of this step

Policy option:	Increase petrol prices by 27.5% on 1 June 2008.
Indicators:	Poverty, defined as household income with a poverty line of IDR42,500/person and week.
	Deforestation, defined as area that cannot be logged in ha.

3. SYSTEM REPRESENTATION

Policy options define the input side of the agent-based model and determine required model features in the Graphical User Interface (GUI). The indicators need to be captured by model outputs as time series data, graphs, or maps. Having these input and output elements for the model allows defining the system boundaries for the model development.

Example for output of this step
(*The diagram has to be defined from the context of each case study and cannot simply be copied from this example*.)



System boundaries specify the spatial and conceptual extent of the model. If, for instance, floods are an important indicator, the system boundaries for hydrological functions are the relevant catchments. While bio-physical indicators often allow for clear spatial boundaries there can be difficulties in the identification of socio-economic boundaries. Especially in times of globalisation many variables are globally driven. Here it is advisable to be guided by the administrative boundaries of decision makers.

The concept of developing a series of agent-based models is partly based on the idea of capturing Indonesia's diversity. If districts in close proximity are extremely similar a smaller (representative) model can be developed for just one of the districts. If districts in close proximity are very different one might want to include them in the model as conclusions from results for one area cannot be made for another area.

The system diagram is developed from the policy options and indicators identified in the first steps of the participatory process. Specify for each of these indicators the variables that determine their state (incoming arrows). These 'explaining variables' are often very context specific, hence the need to specify the spatial boundaries of your work upfront. For instance, poverty of a household in Kutai Barat can depend on the availability of jobs in logging, mining and plantation, and the availability of fish, timer and non-timber forest products. Additionally, poverty is determined by living costs (i.e. food, housing, energy, petrol).

The modeller then needs to identify what variables determine the state of each explaining variable until all links are closed. Many variables will refer to each other. The final list will also include the policy interventions that specify the scenarios. Otherwise the policy interventions would have no impact. The development of a system representation is often done in a system diagram with boxes and links between boxes (see example above). Arrows indicate if a relationship included feedbacks or if it is a one way relationship. Other methods include spreadsheets and UML diagrams.

A ‘stable’ representation is achieved when experts agree with the design. At this stage it is important to start thinking about a reasonable level of aggregation. For instance, one variable might be available jobs in a specific region. Such highly aggregated definitions can be sufficient for many types of research questions. Other problems might require a higher resolution with variables divided into different types of employment (i.e. logging, mining, plantation, other). The process of finding the right level of aggregation goes often through several iterations and is often constraint by data availability.

Next, identify in the systems diagram what ‘boxes’ should change their state endogenously (variables) and which ones should change exogenously (parameters). All variables are *entities*. For each entity identify the relevant attributes the model needs to quantify to describe the entity properly. For instance, households need for the SimPaSI model several attributes, including

- Number of household members
- Household income
- Livelihood(s)
- Location, i.e. ‘Desa’ name

Depending on the context a list of attributes can be very short or very long. You will see that by developing such attribute lists you repeat the conceptualisation: If, for instance, attributes appear for one entity, it means that they are variables the model has to capture. Some of these attributes are shared by multiple entities such as location (i.e. Desa). This means spatial entities (i.e. Desa, Kecamatan, Kabupaten/Kota) have to be considered in this step.

The final list of entities and their attributes should include all indicators the model has to report on as well as all scenario dimensions. Try to keep the system description as simple as possible. This does not mean to take entities or attributes out that are important (significant) for explaining relevant attributes. The definition of attributes includes the definition of scale.

4. DATA FOR INITIATION OF ATTRIBUTES

Technically, all attributes need to be quantified or specified for the start of the model run. Remember that those attributes that remain unchanged are parameters, while those that change endogenously are variables. This means that, with defining the attributes, the state is specified for each entity that is assumed to be a realistic starting point for day 0. The specification can be a number (IDR 100,000) or a qualitative description (high). All attributes for all entities have to have a starting state. Spatial entities require partly GIS data, such as land-cover data for the polygons describing the landscape. The data collection process should start as early as possible.

During the step of deciding what *type* of variable the attribute is. If the entity has to be defined in very specific quantitative terms a number is required. This translates into a so called *Double* or an *Integer* variable. *Double* means the number is calculated with decimals while *Integer* means that they are full numbers only. If the state is defined by a word the variable is a *String*. Such qualitative categorisation of states, such as *high* and *low*, needs to be reflected in the transition rule (or response function) of the attribute. Examples will be given in the section on Pseudo code.

In principle, data has to be collected for five system dimensions, the landscape, the environment, the market, the government and the human agents. Landscape data should entail at least the following five data sets: digital elevation model (DEM), land use data, administrative boundaries (desa, kecamatan, kabupaten), rainfall (as many gauge stations as available for the case study region), and soil data. All datasets should be as disaggregated as possible.

Environmental data depends on the entities entailed in the system diagram. Governmental data points are described by the policy options that define the scenarios that need to be run. Requirements for market data (or economic data) results from the systems diagram, for instance prices for specific commodities or wages.

Data on humans and human behaviour are a domain that is not easily available. In most cases field work has to be conducted to elicit such data. Many methods exist for this step: Surveys, interviews, census data, experiments, participant observation, role-playing games, time series data and expert knowledge. Surveys and interviews are the most common approach for gathering behavioural data. In principle, three sections can be distinguished: Firstly, questions on agent attributes that are relevant for the design of artificial agents. This category entails number of household members, household livelihoods, household income, and education. The complete list of required attributes depends on the modelling context and on the broader methodology applied. This first part can normally be completed in a survey.

The second part of data is behavioural data. One approach is to list one by one the scenario definitions and ask households how they would change relevant variables (i.e. level of livelihood activities) under each condition. Such what-if style questions are normally conducted in interviews with partly open-ended questions. In closed questions the interviewee is given a selection of pre-defined answers. Open-ended questions have no pre-listed options, such as *Yes* or *No*.

Example for output of this step

Closed question:	How many family members are there in your household?
	Choose from the following: ☐1 ☐2 ☐3 ☒4 ☐5 ☐6 ☐more
Open question:	What do you like about trees? ____________________

The third part is optional and can cover additional points, mostly in open-ended questions. This can cover information that allows the modeller to cross-check if information given in the behavioural section is plausible or not.

It is strongly recommended to cooperate with scientists from local universities that are experienced in conducting this type of field work. These experts should already be consulted during earlier stages of the interview development. Often time is needed to train university staff for conducting interviews.

5. RESPONSE FUNCTIONS

Dynamic modelling means that variable states can change. Such changes depend on the state of explaining variables; a functional relationship which is captured in so-called transition rules or response functions. Non-technically speaking, how does an attribute (i.e. household income) change in response to the determining variables? When developing the system representation the set of response functions was already partly defined. The outcome of the system representation is, for instance:

- Household income is a function of natural resource use, natural resource prices, wages, and expenditure.

- More technically written
  householdIncome = f(NaturalResourceUse, NaturalResourcePrice, LabourWages, householdExpenditure)

Response functions specify the relationship between the explaining variables. For instance:

- householdIncome = NaturalResourceUse * NaturalResourcePrice
  + LabourWages – householdExpenditure

If the attribute is defined as an *integer* or a *double* variable the definition can be in such a mathematical function. If the attribute is defined as a *string* type the definition requires a different approach, for instance:

- IF NaturalResourceUse * NaturalResourcePrice
  + LabourWages – householdExpenditure ≥ 100,000

  THEN householdIncome = high

  IF 100,000 > NaturalResourceUse * NaturalResourcePrice
  + LabourWages – householdExpenditure ≥ 30,000

  THEN householdIncome = medium

  ELSE householdIncome = low

Such a qualitative classification means that either quantitative explaining variables are put into ranges that represent a qualitative state (high, medium, low) or that specific combinations of qualitative explaining variables are specified to determine the state of this *string* variable. This description above already builds on the methodology of so-called pseudo code, explained in the next section.

Response functions should be developed by experts. Ecological variables should involve ecologists, hydrological variables hydrologists, etc. Social scientists should be involved for behavioural response functions of individuals and households. All of such response functions are very likely to be context specific. In some cases, such as hydrology, universal laws can be applied and already existing algorithms can be implemented. In most other cases contextual data has to be found. For ecological variables this is potentially do-able in a desktop analysis. For behavioural data this is rarely possible. Developing such data from the field can happen in various ways.

The most common approach for developing response functions is to define a representative sample (using an adequate sample size and an effective stratification strategy). The attributes of the entity *individual* and the entity *household* should then be mapped into a survey instrument. The policy scenarios are then translated into questions regarding how such a change is likely to affect the state of relevant attributes, such as livelihood. After the field work is completed and the database is checked for consistency it can be used for direct up-scaling to initialise the attributes and behavioural response functions of the whole population.

Direct up-scaling involves high levels of uncertainty if the population is large and diverse. Many natural resource linked questions are very sensitive to small groups. Such small groups might respond with increased natural resource, which can have large impacts (externalities) on the rest of the population. If representatives of such small groups are captured in the sample direct up-scaling can create a problem: Direct up-scaling is likely to extend the size of such groups beyond realistic numbers. If the stratification of the survey leads to not capturing one of such small groups, the model will not include their important behaviour. In other words, direct up-scaling is likely to over or under-estimate such important groups because the proportions of behaviours captured in the sample remain for the whole model population.

Reducing uncertainty can be achieved by disproportional up-scaling. Such approaches develop often so-called typologies. Two main methods can be identified and both require non-sample data for the up-scaling process:

- A survey can be conducted that includes questions for behavioural response functions. Statistical clustering (or grouping) method are then applied using the behavioural data section. Each typology needs then to be profiled (i.e. by multi-variate approaches) based on the non-behavioural data section (household characteristics such as education, income). Non-sample data such as census data can then be used to map behavioural typologies into the whole population. This results in up-scaling of behavioural assumptions from the survey sample to the whole population disproportional to the initial sample.

- A survey can be conducted just for non-behavioural data. Clustering (or grouping) methods can then be applied. Resulting typologies need to be profiled, which needs to involve the identification of variables with the highest discriminatory power. Based on these characteristics of clusters surveys or interviews need to be conducted to elicit the behavioural data. During this step interviews will need to be focused on persons that are core representatives of each cluster. This means that interviews need as an entry question the previously identified variables. If a person does not fit a cluster the interview does not need to be carried out. Then the interview data needs to be developed into behavioural response functions for each type. The behavioural response functions will need to be mapped into the whole population by using the typology profiles and census data.

If census data is not available proportional up-scaling becomes the most likely option.

6. PSEUDO CODE

Pseudo code defines the main material for the design document. A design document lists specifications that are necessary for the development of new software or a new agent-based model. The central part of a design document defines model processes in so-called pseudo code. Pseudo code includes the definition of variables, their states and the response functions that specify how states change for each variable. Pseudo code is a structure to define algorithms.

Pseudo code can be accompanied by the specification of the model architecture. This depends on the agreement with the coder who implements the model design. If, for instance, the system representation is developed as a UML the architecture is implicitly suggested with entities defining classes and for each class the list of existing attributes and where linkages are. In many cases a design document specifies pseudo code but leaves it to the coder what explicit architecture is realised.

The advantage of pseudo code is that even a non-technical reader is able to reconstruct the functionality of the model, which increases the transparency. At the same time it specifies the model design to a degree that any programmer can realise the exact purpose of the model designer. Pseudo code guides exist online, for instance:

- http://users.csc.calpoly.edu/~jdalbey/SWE/pdl_std.html

Model implementation refers to the process of implementing the design that is captured as pseudo code in the design document. Normally, this phase requires several iterations involving clarifications of pseudo code and revision of initialisation values. The first steps of model implementation include constant model testing regarding functionality, which reveals often mistakes (i.e. missing attributes, wrong variable types, or impossible parameter values).

Part of this phase is also the development of a GUI. This should be tested in isolation from model performance to ensure that the GUI reflects user needs and is user friendly.

7. IMPLEMENTATION AND SOFTWARE

The modeller documents all pseudo code and background material in a design document; see for examples Smajgl et al.(2009a) and Smajgl et al. (2009b). The design document provides the necessary transparency for non-modellers and allows for effective communication between the modeller and the coder. Once a coder is selected some time should be assigned for explaining all elements of the pseudo code. The coder will implement the pseudo code in a language such as Java or C#. The implementation is conducted in a software development platform such as

Eclipse or NetBeans. The agreement between the model developer and the stakeholders can involve the return of a software product (executable stand alone version) or the return of source code (run through software such as Eclipse or NetBeans). The modeller should be available for clarifications during the implementation. The coder should be available for providing updates during testing and validation.

8. MODEL TESTING AND VALIDATION

After model implementation, the model functionality should be tested. The model needs to be run for each GUI element to test if indicator responses are plausible. Experts can help judge if the actual degree of impact caused by changes on the GUI is ‘realistic’. However, such plausibility should only be tested for primary impacts, which means that only direct links between variables are tested. If more indirect connections, such as those between poverty and petrol prices, are tested, model functions should not be adjusted to fit the belief of an expert. If testing reveals problems the modeller needs to identify possible problems in the pseudo code. Such mistakes should be fixed and the modified design document should be handed over to the coder for adjusting the source code.

After all GUI fields have been tested the model should be validated. Validation describes the process of comparing simulation results with real world data. This means that actual decision should be simulated in an ex-post analysis and compared with official statistics. Important during this step is that the official statistics employs exactly the same definition of the indicators used for the comparison.

Such validation is very important but does not guarantee a good model. A good match of simulation results and real data does not mean that the model can be used for effectively assessing a different policy scenario. Under changing circumstances different mechanisms are likely to play out differently. It is advisable that a lot of energy is placed in validating model *assumptions*. Model assumptions are represented by initialisation values and response functions. Model validity is generally improved by increasingly incorporating real data and expert advice.

9. MODEL ANALYSIS

When stepping into the actual simulation phase the reasonable number of runs to achieve a robust distribution of indicator values has to be determined. Running the same scenario twice will get different results as many values will be defined with a specific distribution. This means that running several hundred simulations will lead to a distribution of results that are unlikely to change after adding more runs. The minimum, the maximum, and the mean will remain effectively unchanged.

Technically, a robust number of runs can be determined in the following approach: A meta file should be created that can read all raw data produced by the model runs. Spreadsheet needs to be created for each indicator with one column per run; on another sheet averages and ranges need to be calculated. Running the model one hundred times produces in the meta file one hundred columns of raw data for each indicator. The modeller can now test how the average changes when stepping from ten runs to eleven, to twelve, etc. The increasing sample will lead to decreasing levels of changes in averages and ranges. When changes become marginal the number of runs is found that can be applied to all other scenarios.

A more technical way to identify the necessary number of runs is to apply bootstrapping methods. For this approach data from the meta file should ne loaded into a software product such as STATA or SPSS. A bootstrapping method needs to be applied over each indicator. The modeller needs to define the indicator, the level of correctness required (i.e. <1%) and the data source (meta file). The bootstrapping method selects randomly sub-samples taken from the whole number of runs (i.e. 100). Depending on the cut-off point the sample size will be determined that leads to stable results. In some cases the result might be that 100 is not sufficient.

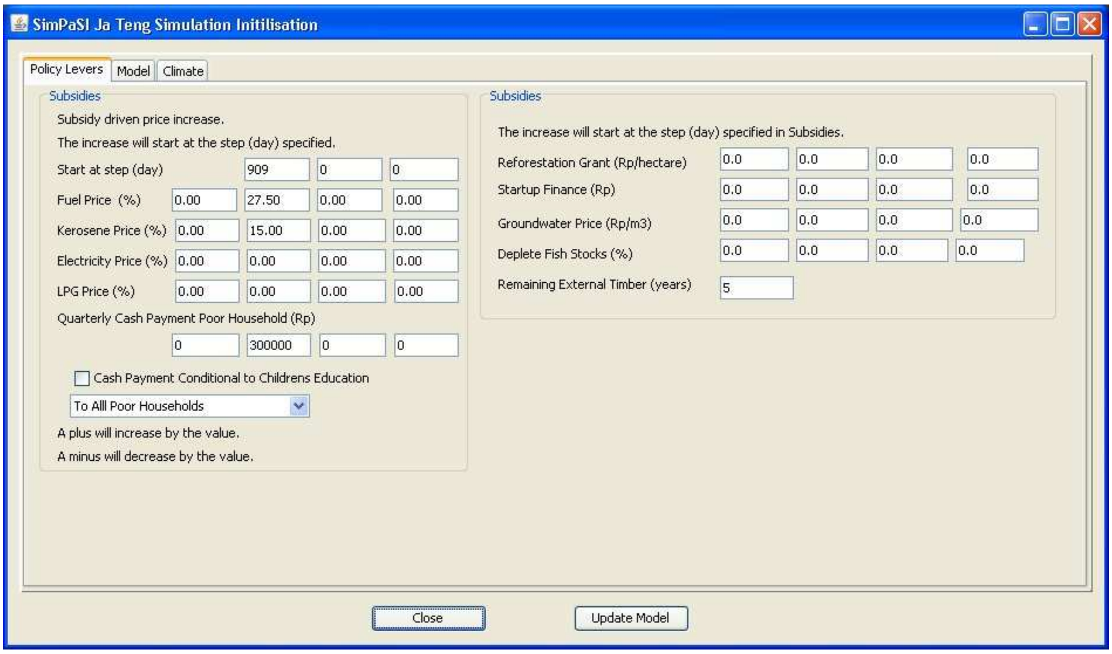


Figure 2: Graphical User Interface (GUI) for SimPaSI Central Java

After the number of runs has been determined each scenario is run accordingly. Analysis of SimPaSI models can be in the style of time series or spatial. This ex-ante time series data analysis allows for applying statistical methods (and software) known from ex-post analysis. The goal is to analyse the policy options identified in the initial step of policy engagement. Most of such policy options are defined in a package of diverse changes. The model should be used to compare such an intervention with a situation without such an intervention (benchmark or base line). Additionally, it is beneficial to simulate the impact of each component of the policy option. It can also provide important insights to test gradual changes of important policy levers. Examples can be found in Smajgl et al. (2009d) and Smajgl et al. (2009c).

Figure 2 shows the GUI for SimPaSI Central Java with central government policy levers to the left and local government levers to the right. An important step in the analysis can be to test the potential impact of combinations of central and local government decisions.

The analysis of time series data implies the development of meaningful charts and numbers and their translation into clear policy messages. It is important to keep in mind that this modelling approach is not seeking precision. Therefore, the main purpose of analysis is to contribute to a discussion between decision makers from multiple tiers of governance. This should be facilitated through workshop situations. Model results can be used to challenge existing beliefs.

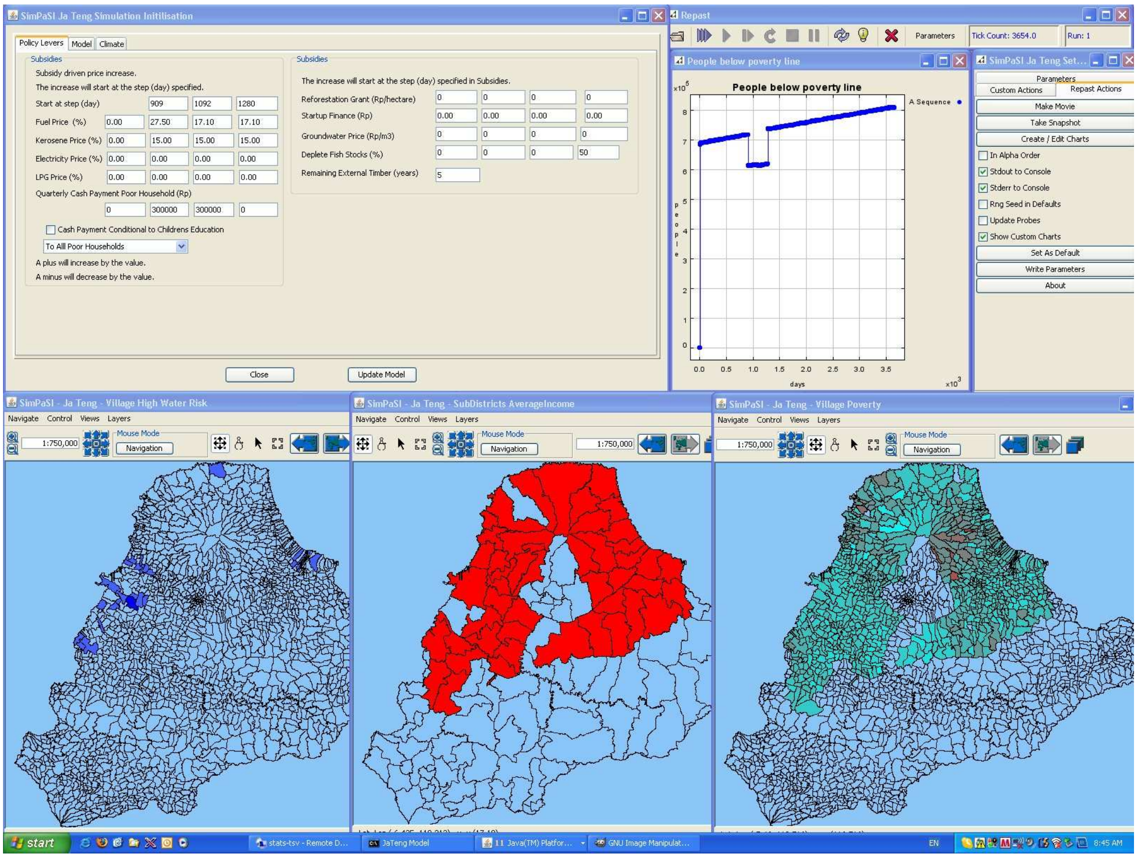


Figure 3: Simulation output for SimPaSI Central Java with maps for (bottom from left) high water risk, average income, and village poverty, and (top from left) GUI, poverty chart, and Repast menu.

Spatial analyses can add value to such an improved understanding of system behaviour. The model needs to produce for all relevant time steps an updated file (dbf type) with spatial references if a spatial analysis is required. For instance, results could document land use changes or poverty developments over time (i.e. annually). Translating such dbf files into updates shape files (maps) can show spatial variations of important indicators and their changes. Maps that visualise geographical areas of large changes allow for the identification and communication of hot spot areas. This can allow decision makers identifying high risk areas for indicators such as poverty fluctuations or high water levels.

The model use is not limited to conducting an analysis and presenting simulation results. Using the model in workshop situations has proven to be very effective to facilitate discussions on effective policy options and their consequences. So-called *live runs* (see Figure 3) can reduce the level of abstraction often present in events of the participatory process. Having stakeholders discussing a model run allows for capturing underlying beliefs these decision makers hold. Documenting them and comparing them can be effective in coordinating decisions across multiple tiers of governance.

REFERENCES

Deadman, P. Modelling individual behaviour and group performance in an intelligent agent-based simulation of the tragedy of the commons. Journal of Environmental Management. 56:159-172; 1999

Gilbert, N. Agent-based models. Los Angeles: SAGE Publications; 2008

Miller, J.H.; Page, S.E. Complex Adaptive Systems: An Introduction to Computational Models of Social Life. Princeton and Oxford: Princeton University Press; 2008

Smajgl, A. Accounting for multiple governance scales in integrated modelling and decision support. In: Proctor W, Van Kerkhoff L, Hatfield Dodds S, eds. Reflecting on integrated mission directed research: Learning from experience in environmental and natural resource management. Collingwood: CSIRO Publishing; 2009

Smajgl, A.; Carlin, G.; House, A.; Kurnia, A.S.; Butler, J.; Sugiyanto, C., et al. Design Document for agent-based model SimPaSI Jawa Tengah. Townsville: CSIRO Sustainable Ecosysyems 2009a

Smajgl, A.; Carlin, G.; House, A.; Pambudhi, F.; Butler, J.; Bohensky, E., et al. Simulating Pathways to Sustainability in Indonesia: Design Document for agent-based model SimPaSI East Kalimantan. Townsville: CSIRO; 2009b

Smajgl, A.; Carlin, G.; Pambudhi, F.; Bohensky, E.; House, A.; Butler, J., et al. Assessing impacts of logging and mining operations on poverty in East Kalimantan, Indonesia: An agent-based analysis. Townsville: CSIRO; 2009c

Smajgl, A.; Carlin, G.; Pambudhi, F.; Bohensky, E.; House, A.; Butler, J., et al. Assessing impacts of fuel subsidy decisions on poverty and forest depletion in East Kalimantan, Indonesia: An agent-based analysis: CSIRO; 2009d

Smajgl, A.; Prananingtyas, S.D. Adaptation dynamics shaped by multiple tiers of governance: Poverty and deforestation in Indonesia. In: Anderssen RS, Braddock RD, Newham LTH, eds. 18th IMACS World Congress ModSim09: International Congress on Modelling and Simulation. Cairns: MSSANZ; 2009

Tesfatsion, L. Agent-Based Computational Economics: Modelling Economies as Complex Adaptive Systems. 2002

Contact Us

Phone: 1300 363 400
+61 3 9545 2176
Email: enquiries@csiro.au
Web: www.csiro.au

Your CSIRO

Australia is founding its future on science and innovation. Its national science agency, CSIRO, is a powerhouse of ideas, technologies and skills for building prosperity, growth, health and sustainability. It serves governments, industries, business and communities across the nation.