Structured Approach to Multi-Paradigm Modeling

Marek Makowski

International Institute for Applied Systems Analysis
A-2361 Laxenburg, Austria

http://www.iiasa.ac.at/~marek
e-mail: marek@iiasa.ac.at

Outline:

- Modeling support for solving complex problems
- Complex systems
- Multi-paradigm modeling
- A structured approach

Modeling support for solving complex problems

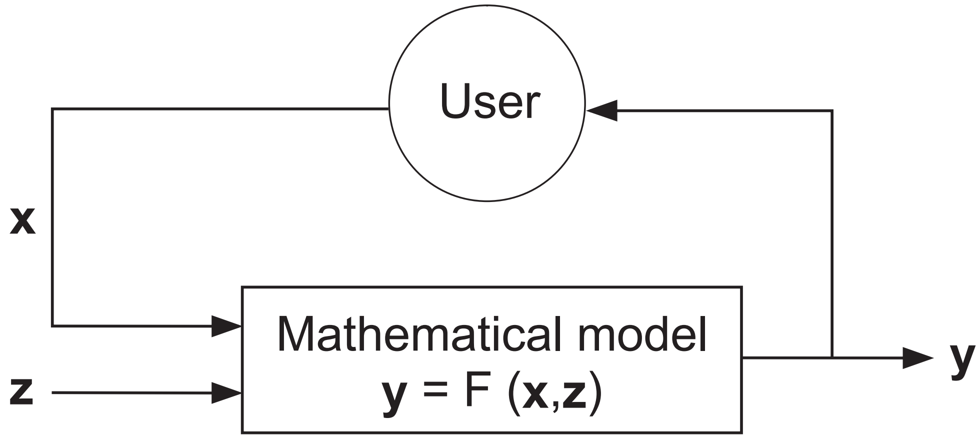

Figure 1: Model-based decision support.

A model-based DS:

- $y = F(x, z)$ can't be handled by intuition or experience
- a rational choice is not that obvious

keep things as simple as possible but not simpler.

No such thing as ...
a simple solution for a complex problem.

Decisions x:
- selection of technologies
- parameters of insurance
- mitigation measures
- ...

External decisions z:
- commitments
- regulations
- subsidy for deductibles
- mitigation measures
- ...

Criteria (objectives) y:
- costs
- various measures of environment quality
- stability of insurance
- ...

Typical complex problem characteristics:
- Infinite number of alternatives
- No transparent aggregation of objectives

Which $y = F(x, z)$?

- habitual domains of modelers and users
- purpose of modeling
- availability of data
- understanding of relations
- acceptability of (local) approximations
- required, desired, feasible accuracy
- transparency of assumptions and specification

Rationality ?

- Optimal is the best ?
  (inverse problem, *not defined* goal(s))
- How decisions are made vs how *we think* should be made
- Bounded rationality
- Rationality vs trade-offs

No single approach to model specification and analysis is adequate for complex problems.

Modeling paradigms

A *modeling paradigm* embodies the consensus of a scientific community on development and analysis of a model, and consists of the theories, rules, concepts, and definitions that are generally accepted in science and practice, as well as of corresponding modeling tools.

T.S. Kuhn, The Structure of Scientific Revolutions, 1974

Model specification paradigms:

- deterministic (also based on statistics)
- stochastic
- fuzzy, fuzzy rules generation
- agent based
- ANN

Model analysis paradigms:

- simulation
- (classical) optimization
- MCMA (incl. soft and inverse simulations)

Why no single paradigm is good enough:

- No unique and adequate specification
- Optimal vs *the best*
- Pareto-optimality
- bounded rationality
- need for learning about the problem
- sovereignty
- ...

Tailoring the problem to a paradigm.

Mathematicians are like Frenchmen:
whenever you say something to them,
they translate it into their own language,
and at once it is something
entirely different.

J.W. v. Goethe

Management Misinformation Systems

The Future of Operational Research is Past

R. L. Ackoff, 1967, 1979

RAINS model

- Used for intergovernmental negotiations,
- Cost effective strategies for improving air quality in Europe,
- Non-lin. model with large linear part (up to 30,000 variables, over 30,000 constraints)

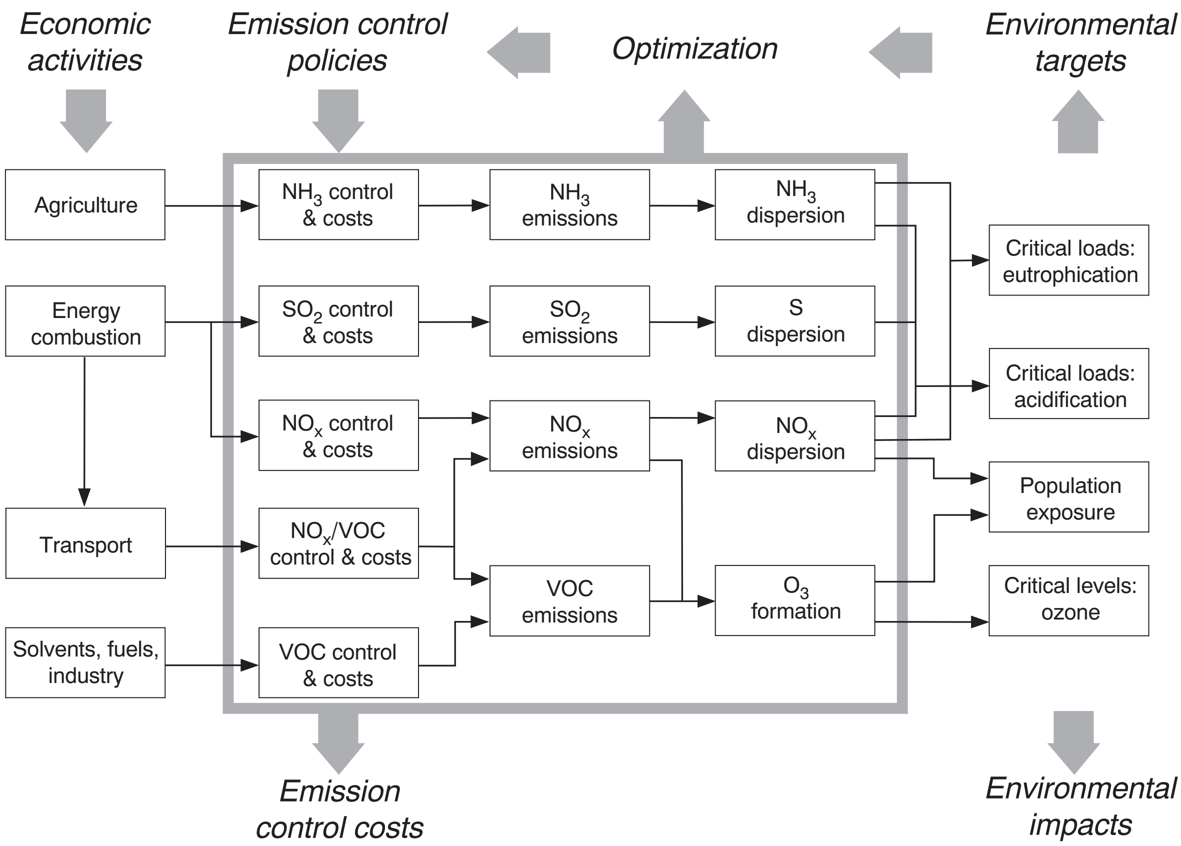

Figure 2: RAINS model structure.

Issues:

- Min. costs to achieve standards,
- Standards are not crisp, PWL for "small" violations,
- Compensations for violating standards,
- Robustness: $c(x_1) \sim c(x_2), \qquad ||x_1 - x_2||$ "large",
- Reference policies.

Methodological/technical problems:

- Smoothing PWL components of the goal function,
- Preprocessing,
- Soft constraints with balances,
- Regularization and soft simulation.

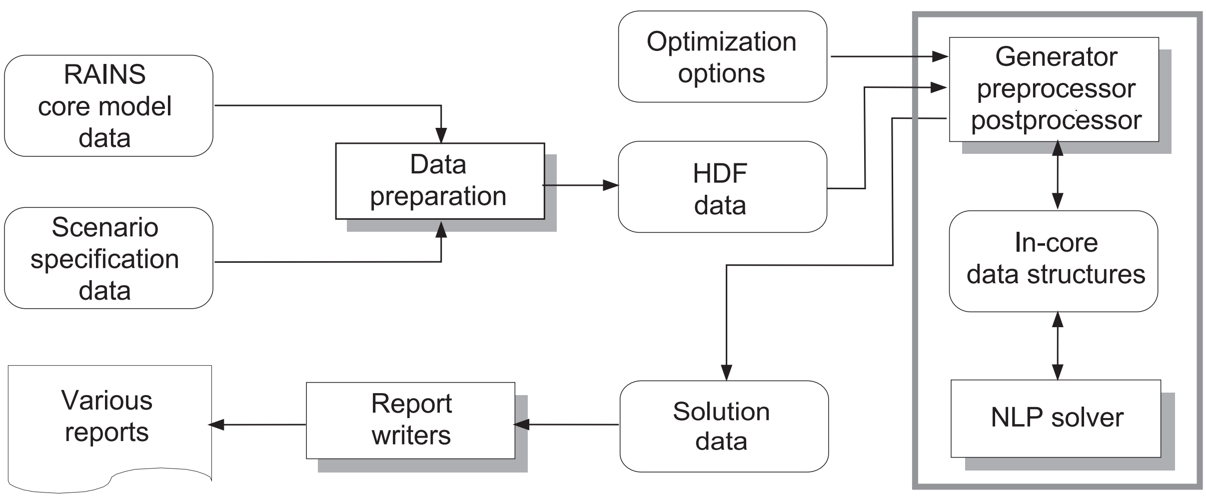

Figure 3: RAINS model analysis cycle.

A structured approach

Basic elements of a modeling cycle:

- Common (user/modeler) problem understanding
- Model (symbolic) specification
- Data for instantiation
- Model instances (specs + data)
- Testing and verification
- Analysis

Documentation for developers, and for users by a (large) team of modelers and users:

- various types of activities (design, development, testing, exploitation)
- analysis of an instance
- comparative analysis of instances
- various requirements for various purposes
- consistency between the documentation and reality

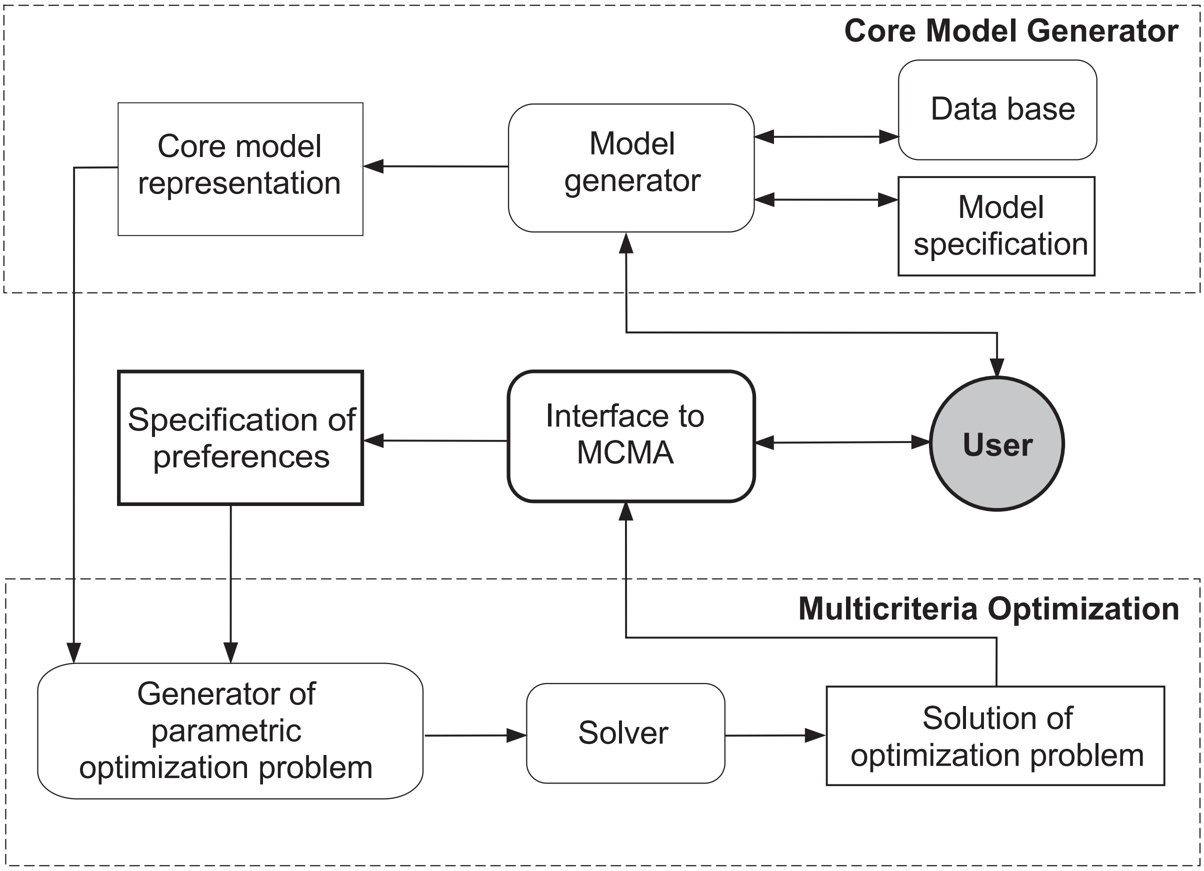

Figure 4: A structure of a modeling cycle for MCMA

Model specification:

- Composed of submodels
- Taxonomy of variables (decision, outcome, auxiliary)
- Check of semantics

Model instance:

- selection of data for various DBs
- documentation of data sources/versions
- completness and consistency of data

Why structured (OO) approach:

- encapsulation of data and methods
- inheritance

Limitations of existing tools:

- Focused on **a** modeling paradigm
- No support for the whole modeling cycle:
  - model specification
  - data handling
  - model verification (incl. semantics)
  - analyses of an instance via various paradigms
  - analyses of various instances
  - maintenance
  - documentation
- No adequate support for coupling models developed indepently

Lessons from the transition of file processing to DBMS:

- separation of data from applications
- standardization (SQL)
- transparency
- portability

Lessons to be applied in advanced modeling:

- separation of the specification (AM) from data (parameters)
- common representation (MSQL ?)
- transparency (one source of the specification)
- all pertinent *views* (analyses)
- distributed resources on heterogeneous hardware

Requirements:

- Support the full cycle (specification, generation, validation, analyses, maintenance, and documentation.
- one source of a specification,
  also help solvers to exploit the model structure.
- Couple model documentation with a version control, history of changes, data used, various views on the model structure, data used, and results of analyses.
- Separate model data, specification and analysis.
- Support for using a model as part of a system of models, and for coupling independently developed models (semantic consistency).
- Support all relevant advanced modeling paradigms (simulation, optimization, multicriteria model analysis, soft simulation).
- Adapt proven DBMS technology, and integrate modeling activities with management of data used for models.
- Support efficient collaborative work.

Is it realistic

DBMS:

- separates data from the applications that use data
- SQL: heterogeneous and distributed resources

Model management as the modeling counterpart to data management (*A. Geoffrion*):
about the same stage of evolution as data management was during its transition from file processing to database processing.

Why it has not been done yet ?

Incremental improvements of fragmented modeling paradigms
vs
a qualitative jump to a new approach to modeling

What is needed ?
Voluntary, de facto standardization of:

- model specification (*long-life MPS*):
  - taxonomy: decision, outcome variables,
  - preferential structure
- (modeling) data management (DBMS)
- model instance
- management of distributed modeling resources

Prototype by P. Celej and P. Rzepakowski

`http://www.iiasa.ac.at/~marek`