Energy Policy 39 (2011) 5850–5870

Contents lists available at ScienceDirect

Energy Policy

journal homepage: www.elsevier.com/locate/enpol

OSeMOSYS: The Open Source Energy Modeling System An introduction to its ethos, structure and development

Mark Howells [a,*], Holger Rogner [b], Neil Strachan [c], Charles Heaps [d], Hillard Huntington [e], Socrates Kypreos [f], Alison Hughes [g], Semida Silveira [a], Joe DeCarolis [h], Morgan Bazillian [i], Alexander Roehrl [j]

[a] Royal Institute of Technology (KTH), Sweden
[b] Planning and Economic Studies Section, International Atomic Energy Agency, Austria
[c] Energy Institute, University College London, UK
[d] Stockholm Environmental Institute (SEI), USA
[e] Stanford University, USA
[f] Paul Scherrer Institute, Switzerland
[g] Energy Research Centre, University of Cape Town, South Africa
[h] North Carolina State University, USA
[i] United Nations Industrial Development Organization (UNIDO), Austria
[j] United Nations Department of Economic and Social Affairs (UNDESA), USA

ARTICLE INFO

Article history:
Received 24 September 2010
Accepted 15 June 2011
Available online 31 July 2011

Keywords:
Energy systems analysis
Energy modeling
Open source

ABSTRACT

This paper discusses the design and development of the Open Source Energy Modeling System (OSeMOSYS). It describes the model's formulation in terms of a 'plain English' description, algebraic formulation, implementation—in terms of its full source code, as well as a detailed description of the model inputs, parameters, and outputs. A key feature of the OSeMOSYS implementation is that it is contained in less than five pages of documented, easily accessible code. Other existing energy system models that do not have this emphasis on compactness and openness makes the barrier to entry by new users much higher, as well as making the addition of innovative new functionality very difficult. The paper begins by describing the rationale for the development of OSeMOSYS and its structure. The current preliminary implementation of the model is then demonstrated for a discrete example. Next, we explain how new development efforts will build on the existing OSeMOSYS codebase. The paper closes with thoughts regarding the organization of the OSeMOSYS community, associated capacity development efforts, and linkages to other open source efforts including adding functionality to the LEAP model.

1. Introduction

1.1. Rationale

OSeMOSYS[1] is designed to fill a gap in the analytical toolbox available to the energy research community and energy planners in developing countries (UK ERC (United Kingdom Energy Research Centre), 2010). At present there exists a useful, but limited set of accessible energy system models. These tools often require significant investment in terms of human resources, training and software purchases in order to apply or further develop them. In addition, their structure is often such that integration with other tools, when possible, can be difficult.

OSeMOSYS is a full-fledged systems optimization model for long-run energy planning. Unlike long established energy systems (partial equilibrium) models (such as MARKAL/TIMES (ETSAP (Energy Technology Systems Analysis Program), 2010), MESSAGE (IAEA (International Atomic Energy Agency), 2010), PRIMES (NTUA (National Technical University of Athens), 2010), EFOM (Van der Voort, 1982) and POLES (Enerdata, 2010)), OSeMOSYS potentially requires a less significant learning curve and time commitment to build and operate. Additionally, by not using proprietary software or commercial programming languages and solvers,OSeMOSYS requires no upfront financial investment. These two advantages extend the availability of energy modeling to the communities of students, business analysts, government specialists, and developing country energy researchers.

Enabling graduate students to build and iteratively develop formal energy models will impart this knowledge base to very wide range of energy market roles and positions. Extending the

* Corresponding author at: Royal Institute of Technology (KTH), Sweden. Tel.: +46 73461 8352.
E-mail address: mark.i.howells@gmail.com (M. Howells).

[1] This effort is based on proof of concept work presented at the International Energy Workshop (IEW) (Howells et al., 2008). That work demonstrated the potential for such an effort focusing on objectives summarized in the acronym "SOFT" (Simple, Open, Free and Transportable).

0301-4215/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.enpol.2011.06.033

human capacity of private and public policy makers to use and understand energy models is a key step in the effective use and interpretation of formal analytical tools (Strachan and Pye, 2009). And growing human capacity in energy modeling in developing countries – whose institutions have relatively fewer research resources – is particularly important, given the growth of developing countries in energy related emissions, resource use, and demand for energy services. However at the most recent International Energy Workshop – the preeminent international energy modeling conference – held in Stockholm in June 2010, less than 10% of participants were from developing countries.

For experienced energy researchers, the OSeMOSYS code is relatively straightforward, elegant and transparent and allows for simple refinements and the ability to conduct sophisticated new analyses. As modeling is designed to generate insights (Huntington et al., 1982), OSeMOSYS allows a test-bed for new energy model developments.

1.2. Paper outline

This paper is divided into two parts. Part 1 discusses the structure of OSeMOSYS, including its division into functional component-'blocks', each representing different levels of abstraction. In part 2, a simple application of OSEMOSYS is presented to provide verification of the model formulation. It goes on to indicate how model development take places by adding a functional 'block' – within each level of abstraction – to the model. Finally, capacity building efforts, including a description of a link to the energy model LEAP, are followed by a description of anticipated next steps.

The paper also includes two appendixes. Consistent with the multiple levels of abstraction used for the model development, including: the algebraic formulation and the implementation in the form of model source code.

2. Part 1: The OSeMOSYS model

OSeMOSYS is designed to be easily updated and modified to suit the needs of a particular analysis. To provide this capability, the model is being developed in a series of component 'blocks' of functionality. A collection of the functional component blocks combines to form a customized model. Further, each block is divided into different levels of abstraction as follows:

I. A plain English description of the model sets, parameters, variables, constraints, and objectives as well as how they are related.
II. An algebraic formulation of the plain English description.
III. The model's implementation in a programming language.
IV. The application of the model, which depends on how it is being used in a study.

Each of these levels is individually important, and by separating them, we hope to facilitate independent and simultaneous additions or refinements.

It is noted that these levels are a simple abstraction of how much model development takes place. However, in existing energy system models, such strata are not neatly identified, nor are they separated. We suggest this lack of separation is conducive to organic and perhaps inefficient model development.

At present the model is disaggregated into seven functional component 'blocks' (Fig. 1). Added together they define the current version of OSeMOSYS. They are compatible and potentially replaceable with new blocks (containing different or improved functionality) with careful and consistent set, variable and parameter definitions. Each block is defined within three levels of abstraction (I–III). The blocks include specifications of the objective (1), costs (2), storage (3), capacity adequacy (4), energy balance (5), constraints (6) and emissions (7).

A full 'Plain English' description, algebraic formulation, and implementation for each 'block' in the current formulation is given in appendix. Under the section "Model Development", the process of adding the storage block (highlighted in Fig. 1) is described. Next we outline the levels of structure (I–V) in more detail, with selected examples extracted from the appendix.

2.1. I. Plain English description

A clearly articulated 'plain English description' is the first level of OSeMOSYS abstraction and represents good practice for any code documentation (McConnell, 2004). This is particularly useful to help match the policy maker and energy system analyst's expectations—a noted downfall of many modeling efforts (Munson, 2004). The full description for each block of the current model is given below.

The objective (block 1 of Fig. 1) in the current version of OSeMOSYS calculates the lowest net present cost (NPC) of an energy system to meet given demand(s) for energy carriers, energy services, or their proxies. (From this point on, 'energy carriers' will be taken to include 'energy services' and 'proxies'). The system is represented by technologies. Most technologies both use and produce energy

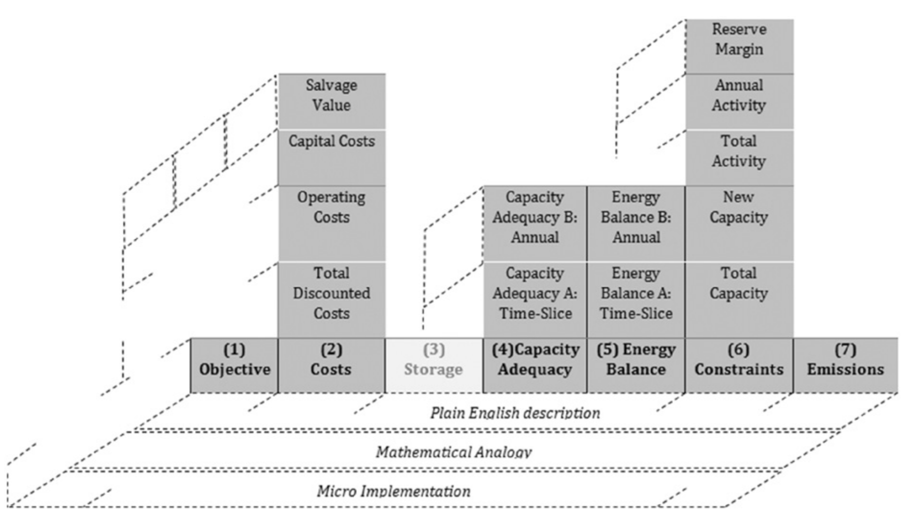

Fig. 1. Current OSeMOSYS 'blocks' and levels of abstraction.

carriers. Energy carrier production must exceed intermediate use by technologies plus any exogenous demand entered by the user.

To meet the objective, there are a number of constraints and specific rules to be followed (Blocks 3–7 in Fig. 1). The full plain English description is given below. Note that for ease of reference selected plain English statements have a bracketed label. These cross reference to the algebraic formulation and micro implementation. In order to illustrate this relationship, one such statement (the constraint given in Block 4: Capacity Adequacy) is referenced and subsequent levels of abstraction discussed.

2.1.1. *Objective (1)*

The objective of this version of the model is to estimate the lowest net present value (NPV) cost of an energy system to meet given demand(s) for energy or energy services.

2.1.2. *Costs (2)*

To do this it should account for the costs incurred by each technology, in each year and in each region modeled. The costs associated with each technology include an operating and investment cost as well as any emission production penalties minus a salvage value. Each cost item should be calculated in constant monetary terms and then discounted to determine a net present value (NPV). To calculate the NPV cost, each technology can have either a default global discount rate or one specific to that technology.

2.1.2.1. Operating costs. The annual operating cost of each technology is the sum of a fixed and variable operating cost. The variable cost is initially calculated for each time slice in a year and the fixed, for each year.

The variable cost is a function of the rate of activity of each technology, its mode of operation and a per-unit cost known by the analyst. For each mode of operation, the rate of activity should be multiplied by the specific per unit operating cost. This is done for each mode of operation in which the technology operates in a time-slice and summed. These costs should then be calculated for each time slice during the year and summed to determine an annual variable operating cost. This is to be done for each technology, each year and each region modeled.

The annual fixed operating cost is calculated by multiplying the total installed capacity of a technology with a per-unit cost known by the analyst. Similarly this should be done for each technology, each year and each region modeled.

The total annual operating cost is the sum of the fixed and variable costs. That cost is then discounted back to the first year modeled. That is done, either using either a technology-specific or global discount rate applied to the middle of the year in which the costs are incurred.

2.1.2.2. Capital cost. Investments (how much of what type of technology when) are calculated on an annual basis, and are assumed to be commissioned and available at the beginning of the year.

The capital costs are determined by the level of new capacity invested in multiplied by a per-unit cost known to the analyst. That cost is then discounted back to the first year modeled. That is done, either using either a technology-specific or global discount rate applied to the beginning of the year in which the technology is available.

2.1.2.3. Salvage cost. When a technology (is invested in during the model period) but ends its operational life before, it is assumed to have no value, at the end of the model period. However, if a technology (invested in during the model period) still has some component of its operational life at the end of the period, that should be estimated. Several methods exist to determine the extent to which a technology has depreciated. And this in turn is used to calculate its salvage value by the end of the period. Sinking fund depreciation is assumed here.

A salvage value is determined, based on the technology's operational life, its year of investment and discount rate. Following this it is discounted to the beginning first model year by a discount rate applied over the modeling period.

2.1.3. *Capacity adequacy (4)*

There must be enough capacity in of a particular technology in order for it to meet its energy use or production requirements. This may need to be the case for each time slice (referred to as Capacity Adequacy "A" below). That is particularly important when considering fuels and services that are difficult or costly to store. There should also enough capacity to meet annual production and use requirements (Referred to as Capacity Adequacy "B" in Box 1 below).

2.1.3.1. Capacity adequacy "A". In order to ensure that there is adequate capacity it is important to establish the total capacity available. Thus the accumulation of all new capacities of all technologies invested during the model period must be calculated for each year (CAa1). To this must be added any residual capacity of the same technology inherited from before the model period. From the addition of the accumulated new capacity and residual capacity in each year of the modeling period, the total annual capacity for each technology is determined. (This is done for each region in the modeling period.) (CAa2).

In order to determine that there is enough capacity, the total rate of activity undertaken by each technology is determined. This involves adding the rate of activity for each mode of operation for each technology. This rate of total activity is determined for each technology in each time slice in each region modeled (CAa3).

All technologies that are expected to have sufficient capacity to meet their activity requirements in each time slice is to be defined by the user. These are then constrained such that their total de-rated capacity is larger than their rate of activity during any time slice. The available capacity at any time-slice is de-rated by a "capacity factor". (CAa3) (Box 1).

2.1.4. *Energy balance (5)*

Operation levels (rate of activity, energy use, energy production and emissions for each mode of operation for each technology) are calculated for "quasi-chronological" "time slices" during the year.

Box 1–A plain English description, cross referenced in subsequent sections.

Capacity Adequacy "B"

All technologies are expected to have a capacity sufficient to account for their annual production (or use) activity. Their annual production is the sum of their production in each time slice. This should be less than their total available capacity multiplied by the fraction of the year for which the technology is available and further de-rated by a "capacity factor" (CAb1).

It is thus important to make sure that the production, use and demand for fuel/energy-services/proxys are feasible at each time slice and annually. (Energy balance “A” accounts for time slice balancing and “B” for annual balancing.)

2.1.4.1. Energy balance “A”. The rate of fuel production for each technology is determined by multiplying the rate of activity to a fuel output/production ratio entered by the analyst. (EBa1) Similarly the rate of fuel usage is determined by multiplying the rate of activity to a fuel output/production ratio entered by the analyst (EBa4). As the technology may be operating in different modes, the rate of production by technology is the sum of production by each mode (EBa2). The same is true for the rate of use (EBa5). For each fuel, time-slice, year and region, the production by each technology can be added to determine the total rate of production of each fuel (EBa4). A parallel approach yields the total use of each fuel (EBa6).

To determine the production (in energy terms) of each fuel (for each time-slice, year and region) the rate of production is multiplied by the length of the time slice (EBa7). Similarly the rate of use (for each time-slice, year and region) is determined (EBa8).

For each fuel there may be an exogenous demand – specific to each time slice (and region) – entered by the analyst. The rate of demand in each time-slice (and each region) is translated to an energy demand by multiplying it by the length of each time-slice (EBa9).

Then for each fuel, in each year, time-slice and region the total production of each fuel should be larger than or equal to its demand and use (EBa10).

2.1.4.2. Energy balance “B”. Summing the production of each fuel for each time-slice during a year gives its annual production (EBb1). Similarly summing the use of each fuel for each time-slice gives its annual use (EBb2). Further, as the analyst may indicate an exogenous accumulated annual demand to be met each year (rather than one specified for each time slice).

Thus for each region and year, another energy balance to be satisfied is that the annual production should be larger than or equal to the annual use and accumulated annual of each fuel (EBb3).

2.1.5. Constraints (6)

There can be a maximum (TCC1) or minimum (TCC2) limit on the total capacity of a particular technology allowed in a particular year and region.

Similarly there is a maximum (NCC1) or minimum (NCC2) new capacity investment limit placed on a particular technology per year and region.

Where specified, a maximum (AAC2) or minimum (AAC3) annual limit on the annual activity of a technology may be entered. The total annual activity of a technology for each year is obtained by adding the product of the rate of activity of each technology with the length of each time-slice during a year for each region (AAC1).

Where specified, a maximum (TAC2) or minimum (TAC3) limit on the model period activity of a technology may be entered. The model period activity of each technology is obtained by summing the total annual activity of each technology for each year for each region (TAC1).

There should be enough capacity (of specified collection of technologies) to provide a reserve margin (for a specified set of fuels) (RM3). By flagging and adding all of the capacities of technologies that are allowed to add to the reserve margin the total capacity in the reserve margin (by year and per region) is determined (RM1). By tagging the fuel and summing them, the demand needing a reserve margin (by year and per region) is determined (RM2).

By summing the production by technology over each time-slice in a year, the annual production by technology (by fuel, technology and region) is obtained (RE1). By flagging which of those technologies are renewable and summing their production, the annual renewable production of a particular fuel (by region) is obtained (RE2).

2.1.6. Emissions accounting (7)

As a technology is active in its various modes of operation, it may impact the environment. The extent to which pollutants are emitted is determined by multiplying a ratio entered by the analyst of “emission per unit of activity” for each mode of operation of a technology. On an annual basis, only the average annual activity by each technology and mode is multiplied by a ratio. (Not all technology will emit pollutants when active, thus only those that are associated with an “emissions per unit of activity” are considered.) (E1).

The annual emissions, from a given technology, are thus determined by summing its annual emissions for each of its modes of operation (E2).

2.1.6.1. Emissions penalty. If there is a per unit penalty associated with the emission, the penalty per emission for each technology is determined. This is achieved by multiplying the annual emissions from each technology with the per unit emission penalty (E3). Summing the penalty for each emission gives the total emissions related penalty associated with the use of a given technology (for each year and model region) (E4).

2.1.6.2. Annual limit. The annual emissions produced for each region are determined by summing the emissions from each technology. The analyst may specify exogenous emissions derived from outside the system modeled. Thus sum of the emissions produced and exogenous emissions gives the annual emissions by emissions species and region modeled (E5). There may be an annual limit placed on emissions by region and emissions species (E7).

2.1.6.3. Model period limit. Summing these emissions for each year and adding exogenous emissions for the model period yields the total emissions over the model period by emission species for each region (E6). (Note exogenous emissions for the model period are independent to those specified on an annual level.)

By region there may also be a limit placed on the model period emissions (E8).

2.2. II. Algebraic formulation

In the full algebraic formulation, variables, sets, parameters, and various operators are described. The algebraic formulation is given in Appendix 1. Each equation is labeled and those labels can be cross referenced with the implementation (model code). A full description of the sets and parameters entered by the analyst to define an application is also described separately, with some level of interpretation under the ‘A data file’ tab of the www.osemosys.org website. In the algebraic formulation, parameters (and variables not determined by the optimization[2]) are indicated in bold.

[2] Note that some variables in the OSeMOSYS formulation are derived and completely specified by parameters, not by meeting the model objective. They are essentially ‘derived parameters’. This allows the analyst to enter data in a familiar format, and improve the readability of the algebraic formulation.

Box 2–The algebraic formulation of 'Capacity Balance' ''B''.

A mathematical interpretation of a power plant constraint (Eq. ((CAb1).

$$\forall_{ytr} \sum_{l} \text{RateOfTotalActivity}_{yltr} \times \textbf{YearSplit}_{\boldsymbol{yl}} \leq \text{TotalCapacityAnnual}_{ytr} \times \textbf{CapacityFactor}_{\boldsymbol{ytr}} \times \textbf{AvailabilityFactor}_{\boldsymbol{ytr}} \times \textbf{CapacityToActivityUnit}_{\boldsymbol{t}}$$

where

l is the intra-annual time step within a year
y is the year modeled
t is the power plant
$\text{RateOfTotalActivity}_{y,l,t,r}$ is the power output
$\text{YearSplit}_{y,l}$ is the fraction of the year
$\text{TotalCapacityAnnual}_{y,t,r}$ is the total installed capacity of the power plant
$\text{CapacityFactor}_{y,t,r}$ is the capacity factor, de-rating the installed to the available capacity
$\text{AvailabilityFactor}_{y,t,r}$ is the availability factor which is $1-M$
M is the fraction of the year in during which planned maintenance takes place
CapacityToActivityUnit relates the unit that capacity is measured in to the unit of activity

Box 3–The implementation (coding) of 'Capacity Adequacy' ''B''.

s.t. CAb1_PlannedMaintenanceEquation{y in Year, t in Technology}: sum{l in TimeSlice} PowerOutput[y,l,t] × YearSplit[y,l] < = TotalCapacityAnnual[y,t] × CapacityFactor[y,t] × AvailabilityFactor[y,t].

It is of particular importance to match the algebraic formulation to the plain English description and document it. It is noteworthy, for example, that while the articulated goals of models such as MARKAL (ETSAP (Energy Technology Systems Analysis Program), 2010) and EFOM (Van der Voort, 1982) were similar, the algebraic formulation and subsequent implementation were significantly different. This helps underscore the utility of coordinated development of the different levels of abstractions. Incidentally, useful aspects of both these divergent models are now combined into TIMES (The Integrated Markal EFOM System (ETSAP (Energy Technology Systems Analysis Program), 2010).

An example of the algebraic formulation of the written example given in the preceding section is in Box 2. Note that long variable and parameter names are chosen and later used in the implementation. The equation (CAb1) is also cross referenced with its word description and the micro implementation. Note also that the equation labeling identifies to which functional 'block' it belongs. 'CAb' is short for 'Capacity Adequacy b' (see Fig. 1) and '1' refers to the equation included in that 'block'. The time dimension in this example is annual, but the code is easily amendable to incorporate other time divisions.

2.3. III. Implementation

From inception to date, the linear programming method and a related programming language called GNU Mathprog (2009) was adopted. This provided several advantages. It allowed a comparison of OSeMOSYS results with the results of similar and widespread tools (e.g., MARKAL/TIMES, MESSAGE)—a useful step to verify the implementation. Mathprog also allows for a transparent implementation of the algebraic formulation. For example, considering the equation CAb1 in the previous section, the implementation (with the prefixed equation reference) is given below. Note that this is done in a significantly more transparent fashion than selected existing tools (see, for example USDOE (United States Department of Energy), 2003) (Box 3).

GNUMathprog[3] is an open source, and freely available, mathematical programming language and represents a subset of the AMPL (2010) language. As such, GNUMathprog bears a strong similarity to proprietary mathematical programming languages such as GAMS (2010), which serves as the basis of the MARKAL model generator (ETSAP (Energy Technology Systems Analysis Program), 2010).

Note, that the user is completely free to employ a different programming language. It is hoped that the explicit algebraic formulation would serve to aid such an exercise.

3. Part 2 OSeMOSYS application

When referring to the application of OSeMOSYS, we are concerned with the manner in which an energy system may be represented given the functional 'blocks' included and applied within the implementation. Application questions, in terms of energy system representation, would include: What should the "reference energy system" (RES) look like[4]? What sets of technologies should be used where? What constraints should be applied? In terms of functional blocks used: Which objective function is to be employed? Is annual or intra-annual time-slice level (or both) balancing required?

A simple demonstration, using the current set of functional 'blocks' is described in the next section. In that example we run OSeMOSYS with the objective of minimizing the total NPV cost of the system (see section I. Plain English Description). That component block (Fig. 1) used to determine this objective may easily be

[3] Please see the "getting started" tab of the www.osemosys.org website for information on how to set about using GNUMathprog.

[4] An early reference to the notion of a RES is given in Bella et al. (1979). Non-standard RES configurations for representing an energy system have been used to model various applications. See Howells et al. (2005) for an example novel RES configuration for modeling appliances and flexible energy service demands in a poor rural village.

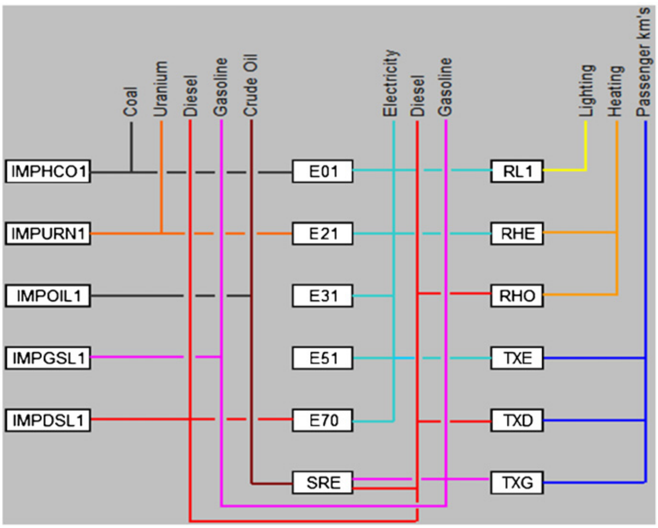

Fig. 2. The UTOPIA reference energy system (RES).

replaced or amended. It is envisioned that future versions of OSeMOSYS will have several potential functional component blocks available. This will include various model "objectives" – for the user to choose from. Anticipating this, it may be useful to think of the example application – and the current version of OSeMOSYS – as running in 'cost minimization mode'. That is deliberately chosen, as many popular tools (MARKAL, MESSAGE, EFOM, WASP), currently run as such by default.

Further, in this version of OSeMOSYS, an energy system is represented by a set of technologies and energy-carriers.[5] Each technology either uses and/or produces an energy carrier. See Fig. 2 for a graphical representation thereof. (Strictly speaking energy is converted from one form to another—it is not 'produced'. The nomenclature adopted is used for convenience). A resource extraction or import process would be described as a technology that produces an energy carrier. An export technology may be represented as a technology that uses an energy carrier. A power plant or refinery represented as a technology that uses an energy carrier, such as crude, and produces an energy carrier, such as electricity, gasoline or diesel. A set of appliances such as lights may be represented as a technology that uses an energy carrier and produces an energy service. Some technologies may operate in more than one 'mode'. A combined heat and power plant (CHP) may, in one 'mode of operation', generate much electricity. In another 'mode of operation' it may generate more heat and less electricity.

Various constraints can be invoked in order to characterize each technology. This is a slight departure from MARKAL and MESSAGE type models that may have different equations generated for technologies in different parts of the energy system (Loulou et al., 2004; IIASA (International Institute of Applied Systems Analysis), 2001). In OSeMOSYS, the same equations are generated for all technologies, but only some of them are invoked based on the data inputted by the user.

At the heart of each technology are two variables. The technology's rate of activity and total installed capacity in each year. The rate at which energy is used by the technology is related to its rate of activity by a ratio given by the user. The rate at which energy is produced is related to its rate of activity by another ratio given by the user. Each unit of activity then relates to a unit of energy used and/or produced. Further, the total rate of activity of each technology is limited by the capacity available. A key condition for each energy carrier is that the production of each should be greater than its (intermediate) use and exogenous demand. Based on these simple relations (and various other constraints) a significant set of energy systems and their constituent processes may be represented. More detail is provided below in the section describing a sample application of OSeMOSYS.

[5] Energy carriers can include energy services as well as proxies—such as 'transport'.

3.1. V. Packaging issues

Clearly defined data input and output is vital to aid tool development and software sustainability. For example, a clearly defined input file facilitates the independent development of an interface, or linkage with an existing interface. While this allows developers to code or amend an interface, it also allows the independent development of the OSeMOSYS code. Indeed, adopting this approach allowed the independent revision of OSeMOSYS and adjustment of the LEAP energy model to link with it as an interface. That effort is briefly discussed in the section on capacity building.

At present, data input for each functional component block is described under the 'A data file' tab on the www.osemosys.org website. It is entered in the form of a text file, in a format determined by GNUMathprog syntax (Makhorin, 2003) and the OSeMOSYS model code.

An advantage of using a structured programming language, such as GNU Mathprog, is its well defined modular structure, interfaces within its 'own' components. For example, its data input, output structures as well as communication with the solver used. By default a freeware solver called the GNU Linear Programming Kit (GLPK) (Maximal, 2010) is used. In addition, GNUMathprog can produce an MPS file (Murtagh, 1981) which can be submitted to another solver. Both inputs and outputs can be linked directly to databases and summary output files generated within the GNUMathprog code.

3.2. UTOPIA

An example application is now made to explain selected functional aspects of the model and the philosophy behind OSeMOSYS representation.

Fig. 2 gives the reference energy system (RES) of the system to be analyzed. All lines represent energy carriers and services, while technologies are represented by white blocks. In the text below, all items relating to OSeMOSYS variables or input parameters are underlined and cross reference with a glossary which is presented in Box 4. (They can also be cross referenced with the algebraic formulation (Appendix 1) and implementation (Appendix 2).)

The example energy system presented is called 'UTOPIA'. It was taken from a standard demonstration application used in MARKAL. Various exogenous demands are given for energy services and the model determines the lowest cost configuration of technology operation and investment to meet those demands.

In the UTOPIA application, a single region is represented has having three demands: lighting, heating, and transport. The lighting and heating demand fluctuate: more lighting is required at night and more heating is required in winter. Lighting is met by the stock of light bulbs (RL1); heating by either electrical (RHE) or oil heaters (RHO); and transport by three different types of vehicles: electric (TXE), diesel (TXD) or gasoline (TXG). To generate electricity, five different power stations are available: coal (E01), nuclear (E21), hydro (E31), pumped-storage (E51) and diesel (E70). Diesel and gasoline are imported (IMPDSL1 and IMPGSL1, respectively) and/or produced by a refinery (SRE) that converts imported crude oil (IMPOIL1). Used only for electricity generation, uranium and coal are also imported (via technologies IMPURN1 and IMPHCO1, respectively).

Box 4–A brief glossary of terms.

Reference energy system (RES): The RES is a graphic abstraction of the energy system that is to be modeled. Typically energy and energy services are indicated by lines and these are connected to 'technologies' in blocks. The energy/energy services and technologies indicated are normally only those that will be examined in detail by the modeler. But, as these are only abstractions, the modeler is free to flexibly represent the system so as to capture the dynamics of interest.

Energy: Energy, energy services or proxies are modeled as required by the analyst. They can be aggregate groups, individual flows or artificially separated, depending on analysis requirements.

Technology: Technology includes any element of the energy system which changes energy from one form to another, uses it or produces it. In a departure from several existing modeling frameworks, all energy system components are set up as a 'technology' in OSeMOSYS. (Thus, a 'resource' is represented as a 'technology' with limits on its activity, for example. It is not a modeled as a separate component in the system. The approach reduces the code needed.). Further, as the model is an abstraction, the modeler is free to interpret the role of a technology at will, where relevant. It may for example represent a single real technologie (such as a power plant) or can represent a heavily aggregated collection of technologies (such as the stock of several million light bulbs), or may even simply be a 'dummy technology', perhaps used for accounting purposes.

Year: OSeMOSYS models the energy system in year-long steps defined by the analyst.

Timeslice: Common to several energy systems models (incl. MESSAGE/MARKAL/TIMES) annual demand is 'sliced' into representative fractions of the year. It is necessary to assess times of the year when demand is high separately from times when demand is low, for fuels that are expensive to store. In order to reduce the computation time, these 'slices' are often grouped. Thus annual demand may be split into aggregate seasons where demand levels are similar (such as 'summer, winter and intermediate'). Those seasons may be subdivided into aggregate 'day types'(such as workdays and weekends), and the day further sub divided (such as into day and night) demand.

Emissions: Any pollutant (or other) emission from the operation of a technology can be account for in OSeMOSYS. Typical examples would include atmospheric emissions such as CO2.

Capacity to activity: As capacity of a technology may be measured in one unit and its energy use or production in another, this parameter relates the two. Thus if – each year – a power plant (with capacity is measured in GW) produces energy (measured in PJ), then this parameter would have a value of 31.536. (That is the potential quantity of PJ's that could be produced by each available GW).

Capital cost: The capital cost of each technology is given as a function of the technology as well as the year in which the technology is invested. As not all technologies (represented in OSeMOSYS) need incur a capacity cost, this is left as zero by default. The units are in currency (such as M$) per unit of capacity (such as GW).

Emission activity ratio: Gives parameter gives the rate of pollutant emitted as a ratio to the rate of a particular mode of activity for a technology.

EmissionsPenalty: EmissionsPenalty is the cost incurred per unit of pollutant emitted. The unit is currency (such as M$) divided by physical quantity (such as Gt).

FixedCost: FixedCost is the cost per unit of capacity for a given technology. As not all technologies (as represented in OSEMOSYS) incur a fixed cost, this is left as zero by default. The unit is currency (such as M$) per unit of capacity (such as GW).

InputActivityRatio: InputActivityRatiogives the rate of input (use) of fuel as a ratio to the rate of activity of the technology is operating.

OutputActivityRatio: OutputActivityRatio gives the rate of output (production) of fuel as a ratio to the rate of activity of the technology is operating.

OperationalLife: Each technology is given a limited operational lifespan, and that is included in this parameter. The unit is years and needed to compute both the economics and replacement requirements of new technologies.

ResidualCapacity: At the beginning of the modeling period typically there is a residual or existing stock of technologies in the system. This parameter represents that capacity (Unlike MESSAGE, LEAP and several other models, it is not calculated based on historical investments and a life associated with those investments. The reason for the departure is that often the life of historical investments may have changed from the time of design or implementation.)

SpecitifiedAnnualDemand: Certain energy-services, fuels or proxys have demands that vary significantly between time-slices. When time-slice level modeling is required, their annual demand is entered (in units of energy such as PJ/a) in this parameter. The profile of demand through the year is entered separately.

SpecifiedAnnualDemandProfile: For each energy-service, fuel or proxy which is given a SpecifiedAnnualDemand, a demand profile is assigned. This parameter indicates the fraction of annual energy-service/fuel/proxy demand that is required in each time step. For each year this should sum to one.

TechWithCapacityRequiredNeededToMeetPeakTS: All technologies who's capacity should be such that they can contribute to meet the peak demand for the fuel that they produce (or use) are flagged. If the parameter is flagged, it has the value 1. If only a fraction of the technology's capacity available to meet peak demand, which may be the case due to intermittency for example, then that fraction is entered into this parameter.

TotalTEchnologyAnnualActivityUpperLimit: Should there be a maximum quantity of activity allowed by a technology over a year, this parameter is invoked. Consider a technology used to represent the extraction of a renewable resource, such as harvesting wood. There may be a maximum annual limit (based on available land). If so, then value of this parameter would correspond accordingly.

TotalTechnologyModelPeriodActivityLimit: Should there be a maximum level of activity by a technology over the whole model period, this parameter is used. The parameter may be invoked for a technology that is simulating the extraction of a finite fossil resource, for example.

VariableCost: The variable cost is the cost per unit of activity of a specified technology

YearSplit: This gives the fraction of the year accounted for in each time slice modeled. The sum of each entry over the year should equal one. Consider, for example, a model constructed with a year that has two timeslices, ''Summer'' and ''Winter'' and those are the same length. Then the YearSplit parameter for each timeslice in that year will be 0.5.

AccumulatedAnnualDemand: This parameter, which can be specified for any energy-service, fuel or proxy modeled, indicates the total exogenous demand of each fuel that must be met for each model year. As the name implies this demand can be met during any, or any number, timeslices.

AnnualTechnologyEmission: This is a variable and calculating for each technology it annual emission by emission species.

Activity: In OSeMOSYS when a technology is operating (producing or using a fuel) it is considered 'active' and its 'activity' variable is 'non-zero'.

NewCapacity: This variable represents the investment of new capacity of each technology. The level of investment is set such that there is enough capacity of technologies to produce fuel/energy-service/proxy requirements.

RateOfActivity: This variable represents the rate of activity for each technology for each year and timeslice defined.

RateOfProductionByTechnology: Based on the 'rate of activity' of a technology the production of a given fuel/energy-service/proxy for each technology is calculated in this variable.

RateOfUseByTechnology: Based on the 'rate of activity' of a technology the use of a given fuel/energy-service/proxy for each technology is calculated in this variable.

TotalCapacityAnnual: Based on the sum of new capacity, minus capacity retirements plus residual capacity, this variable calculates the annual capacity of each technology in the system.

To build the UTOPIA application in OSeMOSYS, there are effectively two key technology variables to which other variables and parameters relate: the rate of activity and capacity. The rate of production of an energy-carrier is related to the producing technology's rate of activity by a simple output to activity ratio. If a technology has a non-zero output to activity ratio for a given energy-carrier, when the technology is active it produces that energy-carrier. Matching logic applies to the rate of use of an energy-carrier. If the input to activity ratio is non-zero for a particular energy carrier, it is used when the technology is active.

Thus, to represent the UTOPIA application in OSeMOSYS each existing and future technology is entered in the input file as a set. For a technology that is to produce an energy-carrier, a non-zero output to activity ratio is added (specifically relating one to the other). Similarly, for a technology that uses an energy-carrier a non-zero 'input to activity ratio' is added.

In OSeMOSYS, costs are incurred when the technology is: (1) active (as the product of a its activity and a given variable cost), (2) when it has non-zero capacity (as the product of its capacity and a given fixed cost), (3) when it is invested in (as the product of new investment and a given capital cost) and (4) when it emits pollutants (as the product of its emissions and a given emissions penalty. (Where emissions are calculated as the product of a technology activity and a given emissions activity ratio.) As the variable (and emissions) costs are related the technology's activity, care must be taken to choose what the activity should represent.

Typically the rate of activity is chosen to represent either the rate of fuel use or output. That in turn determines how the activity to output, input and emissions ratios (and their related costs) are calibrated by the analyst. Further, a technology's capacity is determined in order to sufficiently limit its activity. Thus the definition of capacity of a technology is related to what the activity represents, as either input or output.

For example, a power plant's performance characteristics are typically related to its electrical power output. The user may define the rate of activity of the plant to be its electrical power output. In this case the electrical output activity ratio would be given as 1, making the power output and rate of activity equivalent. The fuel input activity ratio, given by the user, would then be equal to its heat rate (or the reciprocal of its efficiency). Consequently the capacity-related capital cost and fixed cost will be given per unit of power output. For the UTOPIA power plants, capital and fixed costs are entered in million $ per GW). Similarly, the variable cost will be per unit output. In this example it is given in million $ per PJ. In order to account for the capacity to activity unit difference a conversion of 31.56 is used to relate PJ with GW. All the power plants represented in the UTOPIA are specified by the user as technologies with capacities that are needed to meet the models peak demand time slice.

In the case of the refinery; however, it is convenient that the rate of activity represents the rate of crude oil processing. Thus, all input and output activity ratios, variable and fixed costs are given relative to its input. This is achieved by setting the input activity ratio to 1, and the output activity ratio to the efficiency of production for each product. Unlike the power plants, the capacity needed is not required to meet the peak time slice.

Technologies representing imports in OSeMOSYS are set up to have a single output and no input. Setting an output to activity ratio of 1 for each, leaves the variable costs to represent the price of imported fuel (other costs are left as zero). Note that resource extraction technologies could be represented in a similar fashion. However, the analyst may want to invoke annual extraction or model period limits, as rates of activity would be equivalent to the extraction rate. The latter would be achieved by applying an annual or model period activity limit.

In this example, demand technologies are set up with output activity ratios of 1 and other parameters thus calibrated. In the technology instances where some investment has taken place prior to the modeling period a residual capacity for each year where this historical capacity exists is specified. The total annual capacity of each technology is calculated by adding all new capacity investments, plus any residual capacity minus retirements based on the technology life.

As the use, production, and demand for each energy carrier and service must balance at each intra-annual time period and each year, care must be taken to ensure that there are sufficient production options for each energy-carrier/service/proxy. In the UTOPIA example, the year is split into three representative seasons, winter, intermediate (spring and fall), and summer. In each season there is a representative day and night time slice. Thus there are six time slices in which the production, use, and demand for each energy carrier and service must balance. In OSeMOSYS, an exogenous demand (i.e., a fuel demand not tied to a particular energy service) can be entered for any fuel. A specified annual demand which differs from time slice to time slice is thus entered for lighting and heating and an accumulated annual demand for passenger kilometers is entered.

In the current implementation of OSeMOSYS, any energy system representation such as Utopia can be extended to include: any number of technologies; any number of fuels; any number of emissions; two different types (specified and accumulated annual) exogenous demand for any energy-carrier/service/proxy entered; any number of regions; capacity, variable and fixed costs for all technologies; any number of time-slices in a year; any number of years; limitations to technology output by year (via an availability factor); limitations to technology outputs by time-slice (via a capacity factor); limitless activities per technology; technologies with capacities constrained to meet peak demand (and intermediate use) requirements; constraints on both the annual and model period total activity of the technologies; any number of energy storage facilities; and constraints on both the total annual and new capacity investments by technology and emissions limits for each year and the model period (while also accounting for any exogenous emissions to be accounted). This illustrates the scope of the flexibility of OSeMOSYS, which can be broadened further via the addition of new or altered component blocks (as discussed in the Model Development section).

3.2.1. Data

Adapting the data for OSeMOSYS, the UTOPIA application was run in both MARKAL and OSeMOSYS. Selected results are compared. The adaptation is necessary as 10 year time steps (1990, 2000 and 2010) are modeled in MARKAL, while OSeMOSYS is set up to model investments on an annual basis.

The basic data used to calibrate the UTOPIA application are summarized in Table 1. The time horizon is from 1990 to 2010.

Table 1
UTOPIA input data.

Patameter	Input	Input to activity ratio	Output	Output to activity ratio	Capital Cost			Variable Cost	Fixed Cost	Capacity to activity unit	Availability factor	Capacity factor	Life	Residual Capacity			Capacity limits			
				For year	*1990*	*2000*	*2010*						*For year:*	*1990*	*2000*	*2010*	*For year:*	*1990*	*2000*	*2010*
Unit					M$/GW			M$/PJ	M$/GW				years	GW	GW	GW		GW	GW	GW
Technology																	Lower	0.13	0.13	0.13
E01	Coal	3.13	Electricity	1	1400	–	–	0.3	40	31.54	0.8	1	40	0.5	0.325	0.15	Upper	0.13	0.17	0.21
E21	Uranium	2.50	Electricity	1	5000	–	–	1.5	500	31.54	0.8	1	40	0	0	0				
E31	Hydro	3.13	Electricity	1	3000	–	–	0	75	31.54	0.275	1	100	0.1	0.1	0.1				
E51	Electricity	1.39	Electricity	1	900	–	–	0	30	31.54	0.17	1	100	0.5	0.5	0.5	Upper	3	3	3
E70	Diesel	3.40	Electricity	1	1000	–	–	0	30	31.54	0.8	1	40	0.3	0.25	0.2				
Unit					M$/PJ/a			M$/PJ	M$/PJ/a				years	PJ/a	PJ/a	PJ/a		PJ/a	PJ/a	PJ/a
IMPDSL1		0	Diesel	1	0	–	–	10	0	1	1	1	1	N/A	N/A	N/A				
IMPGSL1		0	Gasoline	1	0	–	–	15	0	1	1	1	1	N/A	N/A	N/A				
IMPHCO1		0	Coal	1	0	–	–	2	0	1	1	1	1	N/A	N/A	N/A				
IMPOIL1		0	Crude	1	0	–	–	8	0	1	1	1	1	N/A	N/A	N/A				
IMPURN1		0	Uranium	1	0	–	–	2	0	1	1	1	1	N/A	N/A	N/A				
RHE	Electricity	1	Heating	1	90	–	–	0	0	1	1	1	30	0	0	0	Upper	0		
RHO	Diesel	1	Heating	1	100	–	–	0	1	1	1	1	30	25	12.5	0				
RL1	Electricity	1	Lighting	1	100	–	–	0	9.4	1	1	1	10	5.6	0	0				
SRE	Crude	1	Diesel	0.7	100	–	–	10		1	1	1	50				Upper	0.1		
			Gasoline	0.3																
TXD	Diesel	4.33	Passenger–kms	1	1044	–	–	0	52	1	1	1	15	0.6	0.2	0				
TXE	Electricity	1.21	Passenger-kms	1	2000	1750	1500	0	100	1	1	1	15	0	0	0	Upper	0	4	10
TXG	Gasoline	4.33	Passenger-kms	1	1044	–	–	0	48	1	1	1	15	4.6	1.5	0				
	For year:	*1990*	*2000*	*2010*	For season:	Intermediate	Summer	Winter												
					For time slice:	Day	Night	Day	Night	Day	Night									
Demands	PJ/a	PJ/a	PJ/a																	
Heating	25.2	37.8	56.7			0.12	0.06	0	0	0.55	0.27									
Lighting	5.6	8.4	12.6			0.15	0.05	0.15	0.05	0.5	1									
Passenger-km	5.2	7.8	11.7																	
Year Split						0.1667	0.0833	0.1667	0.0833	0.1667	0.0833									

The discount rate is assumed to be 5%. The objective is to determine what investments must take place when, and how the total stock of technologies should be operated in order to determine the lowest cost system. The lowest cost is the objective function value, measured in millions of dollars.

3.2.2. Results

When both models are simulated with the UTOPIA input data, key results are very similar. Fig. 3 shows the results for the total installed capacity of the power plants modeled. There is a higher investment in coal fired power plants in MARKAL, most likely due to the storage representation and minor adjustments in input data. In 2000 and 2010, the difference is roughly 1% and 5% of the total installed capacity in the electricity supply system represented. This is offset by a higher investment in storage capacity in OSeMOSYS. Investments in hydro and oil power plants are identical.

Fig. 4 represents capacity information for heating systems. In MARKAL, heat is supplied by oil boilers, while in OSeMOSYS small quantities of heat are supplied via stored electricity discharged from the pumped storage plant.[6]

The total objective function reported in MARKAL was $37 billion and $36 billion was reported in OSeMOSYS, representing a difference of less than 2%. Note all costs are expressed as inflation-adjusted 2000$ dollars and are discounted to the year 2000.[7]

3.3. Model development

A defining characteristic of OSeMOSYS development – it is envisioned – will be its flexibility. An analyst may potentially add to or amend the implementation to test new ideas without having to design a full systems model from scratch. This allows for rapid prototyping using a local copy of the model, which may or may not be folded into a larger development effort. The flexibility in model design will also help provide a platform to undertake high impact research and reduce the effort required by, and resource required for, postgraduate study to make substantial contributions to the field.

Development of OSeMOSYS is to take place at different levels. This includes developing new functional component blocks as well as the levels of abstraction of which they (or the model as a whole) is composed. New or amended 'blocks' or 'strata' may be added to or replace existing ones. Further, as more 'blocks' are developed, new combinations may be used to develop new macro formulations and applications.

Attempting to follow this block and strata approach requires a clear and consistent definition of variables, parameters and sets. Descriptions for the present system are given in online under the 'A Data File' tab of the www.osemosys.org website. Integrating new 'blocks' and levels of abstraction also requires careful thought and testing of various combinations to a set of 'calibrated' case studies.

A radical example of a contribution may include changing the programming language used by replacing Level of Abstraction 3 – the implementation – for all existing OSeMOSYS blocks. (This is a task supported by the plain English description and algebraic formulation.) Another contribution may be changing the block that represents the objective function from one that moves from least cost to increased robustness. This requires a new "plain English" description, new "algebraic formulation as well as additions (or wholesale changes) to the implementation of the code.

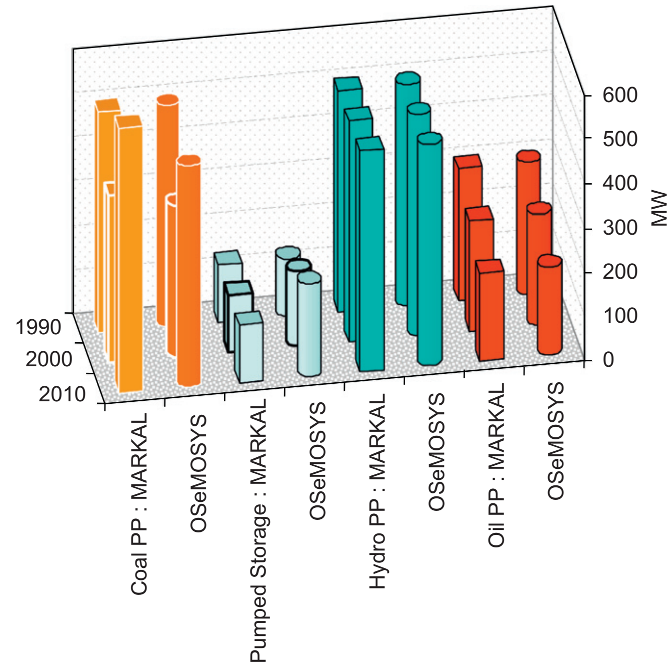

Fig. 3. UTOPIA power plant capacities in MARKAL and OSeMOSYS.

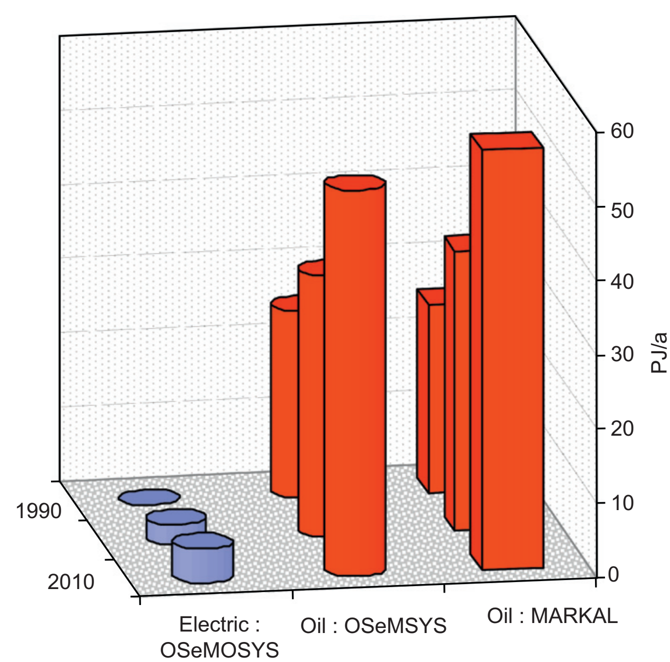

Fig. 4. UTOPIA heater capacities in MARKAL and OSeMOSYS.

3.3.1. The addition of a functional 'block' to the OSeMOSYS model

To illustrate the modularity and flexibility of OSeMOSYS, we describe the addition of the storage facility (see storage block in Fig. 1).[8] Note that OSeMOSYS can and was used to conduct analysis before the inclusion of the storage block. In a similar

[6] From a cursory external investigation, it seems that there should be clear economic scope to include electrode boilers. Electricity produced from a new coal fired plant operating at a load factor of the heat demand (61%) has a levelised cost of 12$/GJ. The full cost of producing heat from a new oil boiler 24$/PJ, compared to 22$/PJ from an electrode boiler using coal based electricity.

[7] By default OSeMOSYS discounts to the first model year, 1990, that was discounted to 2000 outside the model.

[8] While not in its precursor (Howells et al., 2008), this is now incorporated into the current version of OSeMOSYS.

fashion it is envisioned that new or replacement blocks can be developed and added to OSeMOSYS. The subsequent sections are divided by the level of abstraction: 'plain English', algebraic formulation, implementation, and application.

3.4. Plain English description

A storage facility should be charged during the operation of one or more technologies in a specified mode of operation. The ratio of technology activity (of a certain mode) per unit of storage charge rate is known by the analyst. Thus, this ratio multiplied by the rate of activity of a technology operating in a specific mode, multiplied by the duration will give the extent of that charge. The charge is measured in units of energy. The total storage charge is then determined by summing all the technologies and their relevant modes of operation that are specified by the user. (Consider for example a pumped storage station. When its turbine (technology) is operating in charging mode it fills the (storage) dam at a known rate of electricity usage. If there are several turbines (technologies) linked to this dam the cumulative effect of their pumping are used to estimate the storage charge.)

Similarly, a storage facility is discharged during the operation of one or more technologies in a specified mode of operation. The ratio of technology activity (of a certain mode) per unit of storage discharge rate is known by the analyst. Thus, this ratio multiplied by the rate of activity of a technology operating in a specific mode, multiplied by the duration will give the extent of that discharge. (Consider again the pumped storage station. Only in this case the technology is operating in discharge mode. As it is active, the rate at which the storage level drops is a function of a known discharge ratio, determining the rate of electricity generated. Again, if there is more than one turbine discharging water the cumulative effect of these is used to calculate the storage discharge.)

The net storage charge, during any time slice, is the difference between its charge, minus its discharge. With a given start level, the storage level at any time can therefore be determined by adding the net storage charge. The level of the storage, at any time (including any special "inflection times"[9]), should be less than its upper and higher than its lower limit.

Note that the maximum rate of storage charge/discharge is a function of the technologies linked to the storage and related technology (not the storage) parameters. (Again, considering the pumped storage example, the upper (minus the minimum) level of the dam constrains the level of storage possible, while the MW capacity of the turbine (technology) limits the rate at which electricity may be generated.

3.5. Algebraic formulation

$$\forall_{y,t,r}\text{StorageCharge}_{s,y,l,r} = \sum_{t,m}\text{RateOfActivity}_{y,l,t,m,r} \times \textbf{TechnologyToStorage}_{\boldsymbol{t,m,s,r}} \times \textbf{YearSplit}_{\boldsymbol{y,l}} \tag{S1}$$

$$\forall_{y,t,r}\text{StorageDischarge}_{s,y,l,r} = \sum_{t,m}\text{RateOfActivity}_{y,l,t,m,r} \times \textbf{TechnologyToStorage}_{\boldsymbol{t,m,s,r}} \times \textbf{YearSplit}_{\boldsymbol{y,l}} \tag{S2}$$

$$\forall_{s,y,l,r}\text{NetStorageCharge}_{s,y,l,r} = \text{StorageCharge}_{s,y,l,r} - \text{StorageDischarge}_{s,y,l,r} \tag{S3}$$

$$\forall_{b,s,r}\text{StorageLevel}_{b,s,r} = \sum_{l,y}(\text{NetStorageCharge}_{s,y,l,r}/\textbf{YearSplit}_{\boldsymbol{y,l}}) \times \textbf{StorageInflectionTimes}_{\boldsymbol{y,l,b}} \tag{S4}$$

$$\forall_{b,s,r}\text{StorageLowerLimit}_{s,r} \leq \text{StorageLevel}_{b,s,r} \geq \textbf{StorageUpperLimit}_{\boldsymbol{s,r}} \tag{S5, S6}$$

3.6. Implementation

s.t. S1_StorageCharge{s in STORAGE, y in YEAR, l in TIMESLICE, r in REGION}: sum{t in TECHNOLOGY, m in MODE_OF_OPERATION} RateOfActivity[y,l,t,m,r] × TechnologyToStorage[t,m,s,r] × YearSplit[y,l] = StorageCharge[s,y,l,r];
s.t. S2_StorageDischarge{s in STORAGE, y in YEAR, l in TIME SLICE, r in REGION}: sum{t in TECHNOLOGY, m in MODE_OF_OPERATION} RateOfActivity[y,l,t,m,r] × TechnologyFromStorage [t,m,s,r] × YearSplit[y,l] = StorageDischarge[s,y,l,r];
s.t. S3_NetStorageCharge{s in STORAGE, y in YEAR, l in TIME SLICE, r in REGION}: NetStorageCharge[s,y,l,r] = StorageCharge [s,y,l,r] − StorageDischarge[s,y,l,r];
s.t. S4_StorageLevelAtInflection{b in BOUNDARY_INSTANCES, s in STORAGE, r in REGION}: sum{l in TIMESLICE, y in YEAR} Net StorageCharge[s,y,l,r]/YearSplit[y,l] × StorageInflectionTimes[y,l,b] = StorageLevel[s,b,r];
s.t. S5_StorageLowerLimit{b in BOUNDARY_INSTANCES, s in STORAGE, r in REGION}: StorageLevel[s,b,r] > = StorageLower Limit[s,r];
s.t. S6_StorageUpperLimit{b in BOUNDARY_INSTANCES, s in STORAGE, r in REGION}: StorageLevel[s,b,r] < = StorageUpper Limit[s,r].

3.7. Application

In this section we describe how the storage block is added to the other blocks in OSeMOSYS and how it is to be incorporated in an application to represent a component of an energy system.

In order to incorporate a storage facility in an application, there should be a linkage with one or more technologies that charge or discharge the storage technology. This could be used to represent a standard pumped storage system, with a turbine that operates in one mode to charge the storage (using electricity) and in another (generating electricity) to discharge it. It may also be used to represent storage in smart appliances. For example a heater uses electricity and discharges it as heat. A smarter heater may use the electricity when cheap and generate heat for storage, which can be discharged when heat is needed. In OSeMOSYS, the storage would be charged by the heater acting in 'heat generation' mode using electricity. To meet the heating requirement, the discharge of stored energy would be carried out by the heater operating in 'heat dispatch' mode.

To integrate this new block, new variables,[10] parameters[11] and sets[12] were added to the code, and all other variable matched existing definitions. Various new additions to this representation may be made in future iterations. For example, allowing for storage loss as a function of the storage medium's residence time. (Consider for example water evaporating from a dam or, in the latter example, radiative heating loss.) Such additions may be made, documented and assimilated in a relatively transparent manner.

---

[9] 'Inflection times' are times in the year where we may expect the storage level to start changing. For example the last peak-demand-day in the year etc.

[10] *Added variables*: StorageCharge, StorageDischarge, NetStorageCharge and StorageLevel

[11] *Added parameters*: TechnologyToStorage, TechnologyFromStorage, Storage InflectionTimes, StorageUpperLimit and StorageLowerLimit.

[12] *Added set*: STORAGE.

3.8. Capacity building

Much can be done to bolster efforts to build skills and capacity in the area of energy systems modeling. The introduction of OSeMOSYS in teaching programs can be an effective way to disseminate the use of the tool and enhance interest in modeling activities, as well as supporting other energy modeling communities. This could be an important contribution not only to build the capacity to perform systems-level energy planning and analysis, but also to bring additional focus on sustainability to engineering and energy policy curriculums. The formation and bolstering of core communities with modeling capacity helps to ensure the further development of OSeMOSYS as well as other modeling tools. OSeMOSYS provides several distinct advantages for analysts and students:

1. All the components are free and accessible, via the Internet.
2. OSeMOSYS consists of a series of functional components or 'blocks'—there is the potential to amend existing ones or create new blocks with added capability. This allows the energy systems analyst or postgraduate student opportunities to contribute to model development in a demonstrable manner.
3. The focus on ''strata'ed'' development allows analysts to develop additions that focus on a specific area. For example, policy analysts may focus on better describing the challenge and solutions for specific issues being addressed by a decision maker. Energy systems analysts may wish to consider different applications to various energy systems. This ideally allows contributions focusing on key competencies in relevant areas, while still allowing general research and development that may span several development strata.
4. As all underlying components are open source, at any point in the system analysts can build its capacity. This in turn implies that all components of the official system will be available to rigorous and transparent review and scrutiny. (Note that should an analyst take and use components of the code, they are not obliged to return additions, but are encouraged to do so.)
5. OSeMOSYS is deliberately limited and compact, allowing the analyst to develop and ''test drive'' new formulations and ideas relatively quickly. Not only does this allow postgraduate students and emerging analysts ideal thesis material, but also provides a test-bed for the development of existing or new more complex tools.

The open nature of this effort provides an ideal opportunity to contribute to other initiatives. This currently includes linkages with other open source projects, such as the more recent TEMOA project (DeCarolis et al., 2010), as well as existing efforts related to MARKAL/TIMES and MESSAGE. For example, a database schema is currently being designed to store input/output data for both the OSeMOSYS and TEMOA models.

Another ongoing effort in this direction is to link OSeMOSYS to an existing tool called the Long range Energy Alternatives Planning System (LEAP). LEAP is perhaps the most widely used energy systems analysis tool in the world (COMMEND, 2010). It currently focuses on accounting for energy technology capacities and energy flows in one or several regions. LEAP outputs are made through simple projections based on user-specified inputs and growth rates rather than by an optimization algorithm. It is a relatively simple task to addoptimization capability into LEAP by incorporating OSeMOSYS. The figure below shows elements of that development. OSeMOSYS is being linked with LEAP to allow electricity sector expansion to be optimized. While incorporation of optimization adds functionality to LEAP, the well-developed graphical user interface of LEAP also adds significant functionality to OSeMOSYS (Fig. 5).

Further as LEAP is used in many developing countries (COMMEND, 2010) the linkage brings valuable utility to help evaluate policy options identify energy investments needed to underpin much needed development (Bazillian et al., 2010). (More information on the LEAP-OSeMOSYS link can be found in Heaps et al. (2011).)

3.9. Organization of development

The organization of the development of OSeMOSYS s focused around a website: www.osemosys.org . The website is organized by a steering committee.[13] The first iterations of the OSeMOSYS (2009) code were uploaded in 2009, and it has since been updated several times. Successive versions of the code can be found on the www.osemosys.org home page.[14] Further, effort has been made to undertake development in a consultative and open process garnering feedback at several meetings.[15] At present several research efforts are ongoing that both address important modeling questions, provide input for code development as well as streamlining the introduction of OSeMOSYS to new users.[16] OSeMOSYS is also actively being incorporated into teaching programs (KTH (Royal Swedish Institute of Technology), 2009).

Being open source, all contributions and applications are welcome—and invited. Based on reviews of these contributions by the steering committee, voluntary additions will be incorporated into one supported version of the model. It is licensed under the Apache version 2.0 (Apache, 2004).

4. Conclusions and next steps

This paper summarized the structure, an application and aspects of the development of OSeMOSYS, the free and open source energy modeling system. It emphasized its open accessible nature, clear levels of abstraction and the potential for its use and future development. As such this effort, a first of its kind,[17] demonstrated in 2008 (Howells et al., 2008) fills a gap in the existing energy modeling toolkit.

As there may be several interesting and popular additions (as well as corrections) to be incorporated into the core code, it is envisaged there will be a six month release cycle. At this time, a core documented version of the components of the tool will be released.

[13] The current steering committee consists of members from: University College London, Stockholm Environmental Institute, University of Cape Town, North Carolina State University, the Royal Swedish Institute of Technology, Paul Sherrer Institute and the International Atomic Energy Agency.

[14] Note that key changes incorporated, including basic errors, corrections and changes to improve user/developer experience are noted. These include those encountered since the initial code was written and publicly presented in 2008 (Howells et al., 2008). These errors have often picked up by beta testers. Documentation of these changes are included online under the 'Core model evolution' tab at www.osemosys.org.

[15] Including the 2008 (Howells et al., 2008) and 2010 (Howells et al., 2010) International Energy Workshop meetings, side events (KTH, 2010). Other meetings have discussed and demonstrating aspects of the code—these include (UK ERC, 2010), KTH (2011). Note that meeting agenda's and pre-discussions are available online, under the 'Meetings' tab of the www.osemosys.org website.

[16] Descriptions of these efforts can be found online at the www.osemosys.org under the 'Current applications' tab. These include modeling electrification (Welsch et al., 2011), multi-resource modeling (Hermann et al., 2011), Energy security analysis (Howells et al., 2011) as well as integration of OSeMOSYS with LEAP (Heaps et al., 2011).

[17] It is noted that newer efforts (TEMOA, 2011) – with whom there is active collaboration – are being built in a synergistic and complementary fashion. Further, other efforts such as ETSAP (2010) tools allow active participation in their development, however they rely on proprietary software. The widely used tool, MESSAGE, maintains tight and closed development, but it is regularly updated in response to user requests (Strubegger, 2010).

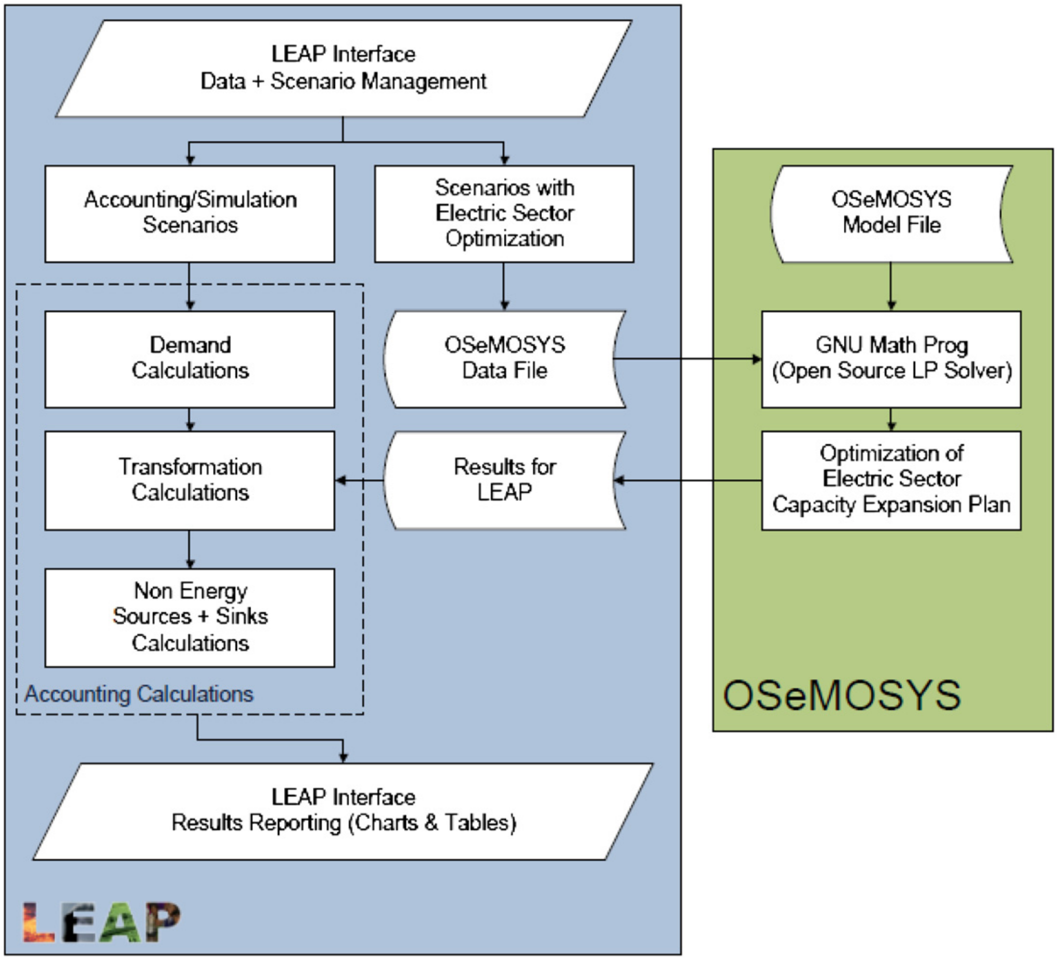

Fig. 5. LEAP and OSeMOSYS.

This will include a set of OSeMOSYS "blocks" at various levels of abstraction that the analyst can use as well as a tested configuration of the tool (such as the Utopia application presented here). In each case, contributors will be required to contribute blocks that include their 'plain English' description, mathematical algebraic formulation, implementation and instructions on their potential application.

Specific areas identified for development include non-linear algorithms as well as representing aspects that have eluded analysis in similar existing tools. These include representing reliability more accurately, smart grids and the long term implications of short term constraints, such as grid stability.

The next steps include the development of an online community, established in such a manner that new development ideas can be submitted, showcased and implemented. This is particularly aimed at encouraging meaningful model development through academic projects (including, for example, postgraduate student input in the form of thesis work). Part of this effort will continue hosting open workshops (such as KTH (Royal Swedish Institute of Technology), 2010).

Initiatives such as the linkage with LEAP as well as other projects will continue. An immediate outcome of this will include the availability of at least one user friendly interface. When considering the widespread use of LEAP in developing countries as well as financial and bureaucratic constraints associated with acquiring many other models, it is envisaged that OSeMOSYS will be a useful tool for building human capacity for energy systems analysis at an applied (as well as development) level.

With an open and costless structure, it may be an ideal candidate solution for development organizations who wish to assist build in-country capacity in both developed and developing country settings. In fact, given the lack of related teaching material on energy systems modeling, an important future effort would be to compile an accompanying text book.

Acknowledgments

This work is indebted to various individuals and organizations. These include, amongst others, the International Atomic Energy Agency (IAEA) whose staff developed the initial concept. The UK Energy Research Centre (UK-ERC) who supported an initial kick-off meeting of experts and contributors in Oxford, February 2010. UNIDO, the United Nations Industrial Development Organization, supported the development of the OSeMOSYS-LEAP Linkage reported. The Stockholm Environmental Institute are kindly hosting an online forum. Individuals who have been particularly supportive of this effort include Manfred Strubegger, the late Leo schrattenholzer, Gary Goldstein and Irej Jalal. Andrii Gritsevkyi needs special mention for his insight and direction. Rodrigo Ceron[18] provided invaluable assistance developing material for and training in the use of the GNUMathprog language.

[18] rceron@gmail.com

Appendix 1. The alegebraic formulation

OBJECTIVE

minimize $\sum_{y,t,r}$ TotalDiscountedCost$_{y,t,r}$ (OBJ)

COSTS

TOTAL DISCOUNTED COSTS

$\forall_{y,t,r}$ TotalDiscountedCost$_{y,t,r}$ = DiscountedOperatingCost$_{y,t,r}$+DiscountedCapitalInvestment$_{y,t,r}$ + DiscountedTechnologyEmissionsPenalty$_{y,t,r}$- DiscountedSalvageValue$_{y,t,r}$ (TDC1)

OPERATING COSTS

$\forall_{y,l,t,r}$ VariableOperatingCost$_{y,l,t,r}$ = $\sum_m$ RateOfActivity$_{y,l,t,m,r}$***VariableCost$_{y,t,m,r}$*** (OC1)

$\forall_{y,t,r}$ AnnualVariableOperatingCost$_{y,t,r}$ = $\sum_l$ VariableOperatingCost$_{y,l,t,r}$ (OC2)

$\forall_{y,t,r}$ AnnualFixedOperatingCost$_{y,t,r}$ = TotalCapacityAnnual$_{y,t,r}$***FixedCost$_{y,t,r}$*** (OC3)

$\forall_{y,t,r}$ OperatingCost$_{y,t,r}$ = AnnualFixedOperatingCost$_{y,t,r}$ + AnnualVariableOperatingCost$_{y,t,r}$ (OC4)

$\forall_{y,t,r}$ DiscountedOperatingCost$_{y,t,}$r = OperatingCost$_{y,t,r}$/((1+**DiscountRate$_{t,r}$**)^(y-**StartYear**+0.5) (OC5)

CAPITAL COSTS

$\forall_{y,t,r}$ CapitalInvestment$_{y,t,r}$ = **CapitalCost$_{y,t,r}$** * NewCapacity$_{y,t,r}$ (CC1)

$\forall_{y,t,r}$ DiscountedCapitalInvestment$_{y,t,r}$ = CapitalInvestment$_{y,t,r}$/((1+**DiscountRate$_{t,r}$**)^(y-**StartYear**)) (CC2)

SALVAGE VALUE

$\forall_{t,r,y:\ (y\ +\ OperationalLifet,r) < StartYear\ +\ card(YEAR)}$ SalvageValue$_{y,t,r}$ = 0 (SV1)

$\forall_{t,r,y:\ (y\ +\ OperationalLifet,r) \geq StartYear\ +\ card(YEAR)}$ SalvageValue$_{y,t,r}$ = NewCapacity$_{y,t,r}$* CapitalCost$_{y,t,r}$*((1-((((1+**DiscountRate$_{t,r}$**)^(**StartYear**+card(**YEAR**) - y))-1) /((1+**DiscountRate$_{t,r}$**)^**OperationalLife$_{t,r}$**-1))) (SV2)

$\forall_{y,t,r}$ DiscountedSalvageValue$_{y,t,r}$ = SalvageValue$_{y,t,r}$/((1+**DiscountRate$_{t,r}$**)^(1+card(**YEAR**))) (SV3)

STORAGE

$\forall_{y,t,r}$ StorageCharge$_{s,y,l,r}$ = $\sum_{t,m}$ RateOfActivity$_{y,l,t,m,r}$ * **TechnologyToStorage$_{t,m,s,r}$** * **YearSplit$_{y,l}$** (S1)

$\forall_{y,t,r}$ StorageDischarge$_{s,y,l,r}$ = $\sum_{t,m}$ RateOfActivity$_{y,l,t,m,r}$ * **TechnologyFromStorage$_{t,m,s,r}$** * **YearSplit$_{y,l}$** (S2)

$\forall_{s,y,l,r}$ NetStorageCharge$_{s,y,l,r}$ = StorageCharge$_{s,y,l,r}$ - StorageDischarge$_{s,y,l,r}$ (S3)

$\forall_{b,s,r}$ StorageLevel$_{b,s,r}$ = $\sum_{l,y}$ (NetStorageCharge$_{s,y,l,r}$/**YearSplit$_{y,l}$**)***StorageInflectionTimes$_{y,l,b}$** (S4)

$\forall_{b,s,r}$ **StorageLowerLimit$_{s,r}$** ≤ StorageLevel$_{b,s,r}$ ≥ **StorageUpperLimit$_{s,r}$** (S5 & S6)

CAPACITY ADEQUACY

CAPACITY ADEQUACY "A"

$\forall_{y,t,r}$ AccumulatedNewCapacity$_{y,t,r}$ =$\sum_{yy:\ y-yy < OperationalLife[t,r]\ \&\&\ y-yy>=0}$ NewCapacity$_{yy,t,r}$ (CAa1)

$\forall_{y,t,r}$ TotalCapacityAnnual$_{y,t,r}$ = AccumulatedNewCapacity$_{y,t,r}$+ **ResidualCapacity$_{y,t,r}$** (CAa2)

$\forall_{y,t,l,r}$ RateOfTotalActivity$_{y,l,t,r}$ = $\sum_m$ RateOfActivity$_{y,l,t,m,r}$ (CAa3)

$\forall_{y,t,l,r:TechWithCapacityNeededToMeetPeakTS[t,r]<>0}$ RateOfTotalActivity$_{y,l,t,r}$ ≤ TotalCapacityAnnual$_{y,t,r}$ * **CapacityFactor$_{y,t,r}$***__CapacityToActivityUnit$_{t,r}$__ (CAa4)

CAPACITY ADEQUACY "B"

$\forall_{y,t,r}$ $\sum_l$ RateOfTotalActivity$_{y,l,t,r}$***YearSplit$_{y,l}$** ≤ TotalCapacityAnnual$_{y,t,r}$***CapacityFactor$_{y,t,r}$*** **AvailabilityFactor$_{y,t,r}$***__CapacityToActivityUnit$_{t,r}$__ (CAb1)

ENERGY BALANCE

ENERGY BALANCE "A"

$\forall_{y,l,f,t,m,r}$ RateOfActivity$_{y,l,t,m,r}$***OutputActivityRatio$_{y,t,f,m,r}$** = RateOfProductionByTechnologyByMode$_{y,l,t,m,f,r}$ (EBa1)

$\forall_{y,l,f,t,r}$ RateOfProductionByTechnology$_{y,l,t,f,r}$ = $\sum_m$ RateOfProductionByTechnologyByMode$_{y,l,t,m,f,r}$ (EBa2)

$\forall_{y,l,f,r}$ RateOfProduction[y,l,f,r] = $\sum_t$ RateOfProductionByTechnology$_{y,l,t,f,r}$ (EBa3)

$\forall_{y,l,f,t,m,r}$ RateOfUseByTechnologyByMode$_{y,l,t,m,f,r}$ = RateOfActivity$_{y,l,t,m,r}$***InputActivityRatio$_{y,t,f,m,r}$** (EBa4)

$\forall_{y,l,f,t,r}$ RateOfUseByTechnology$_{y,l,t,f,r}$ = $\sum_m$ RateOfUseByTechnologyByMode$_{y,l,t,m,f,r}$ (EBa5)

$\forall_{y,l,f,r}$ RateOfUse$_{y,l,f,r}$ = $\sum_t$ RateOfUseByTechnology$_{y,l,t,f,r}$ (EBa6)

$\forall_{y,l,f,r}$ Production$_{y,l,f,r}$ = RateOfProduction$_{y,l,f,r}$***YearSplit$_{y,l}$** (EBa7)

$\forall_{y,l,f,r}$ Use$_{y,l,f,r}$ = RateOfUse$_{y,l,f,r}$***YearSplit$_{y,l}$** (EBa8)

$\forall_{y,l,f,r}$ Demand$_{y,l,f,r}$ = RateOfDemand$_{y,l,f,r}$***YearSplit$_{y,l}$** (EBa9)

$\forall_{y,l,f,r}$ Production$_{y,l,f,r}$ ≥ Demand$_{y,l,f,r}$ + Use$_{y,l,f,r}$ (EBa10)

ENERGY BALANCE "B"

$\forall_{y,f,r}$ $ProductionAnnual_{y,f,r} = \sum_l Production_{y,l,f,r}$ (EBb1)
$\forall_{y,f,r}$ $UseAnnual_{y,f,r} = \sum_l Use_{y,l,f,r}$ (EBb2)
$\forall_{y,f,r}$ $ProductionAnnual_{y,f,r} \geq UseAnnual_{y,f,r} + \mathbf{AccumulatedAnnualDemand_{y,f,r}}$ (EBb3)

ACTIVITY, CAPACITY, RESERVE MARGIN AND RENEWABLE PRODUCTION CONSTRAINTS

TOTAL CAPACITY CONSTRAINTS

$\forall_{y,t,r}$ $TotalCapacityAnnual_{y,t,r} \leq \mathbf{TotalAnnualMaxCapacity_{y,t,r}}$ (TCC1)
$\forall_{y,t,r}$ $TotalCapacityAnnual_{y,t,r} \geq \mathbf{TotalAnnualMinCapacity_{y,t,r}}$ (TCC2)

NEW CAPACITY CONSTRAINTS

$\forall_{y,t,r}$ $NewCapacity_{y,t,r} \leq \mathbf{TotalAnnualMaxCapacityInvestment_{y,t,r}}$ (NCC1)
$\forall_{y,t,r}$ $NewCapacity_{y,t,r} \geq \mathbf{TotalAnnualMinCapacityInvestment_{y,t,r}}$ (NCC2)

ANNUAL ACTIVITY CONSTRAINTS

$\forall_{y,t,r}$ $TotalTechnologyAnnualActivity_{y,t,r} = \sum_l RateOfTotalActivity_{y,l,t,r} * \mathbf{YearSplit_{y,l}}$ (AAC1)
$\forall_{y,t,r}$ $TotalTechnologyAnnualActivity_{y,t,r} \leq \mathbf{TotalTechnologyAnnualActivityUpperLimit_{y,t,r}}$ (AAC2)
$\forall_{y,t,r}$ $TotalTechnologyAnnualActivity_{y,t,r} \geq \mathbf{TotalTechnologyAnnualActivityLowerLimit_{y,t,r}}$ (AAC3)

TOTAL ACTIVITY CONSTRAINT

$\forall_{t,r}$ $TotalTechnologyModelPeriodActivity_{t,r} = \sum_y TotalTechnologyAnnualActivity_{y,t,r}$ (TAC1)
$\forall_{t,r}$ $TotalTechnologyModelPeriodActivity_{t,r} \leq \mathbf{TotalTechnologyModelPeriodActivityUpperLimit_{t,r}}$ (TAC2)
$\forall_{t,r}$ $TotalTechnologyModelPeriodActivity_{t,r} \geq \mathbf{TotalTechnologyModelPeriodActivityLowerLimit_{t,r}}$ (TAC3)

RESERVE MARGIN CONSTRAINT

$\forall_{y,r}$ $TotalCapacityInReserveMargin_{y,r} = \sum_t TotalCapacityAnnual_{y,t,r} * \mathbf{ReserveMarginTagTechnology_{y,t,r}} * \mathbf{CapacityToActivityUnit_{t,r}}$ (RM1)
$\forall_{y,l,r}$ $DemandNeedingReserveMargin_{y,l,r} = \sum_t RateOfDemand_{y,l,f,r} * \mathbf{ReserveMarginTagFuel_{y,f,r}}$ (RM2)
$\forall_{y,l,r}$ $TotalCapacityInReserveMargin_{y,r} \geq DemandNeedingReserveMargin_{y,l,r} * \mathbf{ReserveMargin_{y,r}}$ (RM3)

EMISSIONS ACCOUNTING

$\forall_{y,t,e,m,r}$ $AnnualTechnologyEmissionByMode_{y,t,e,m,r} = \sum_l \mathbf{EmissionActivityRatio_{y,t,e,m,r}} * AverageAnnualTechnologyActivityByMode_{y,t,m,r}$ (E1)
$\forall_{y,t,e,r}$ $AnnualTechnologyEmission_{y,t,e,r} = \sum_m AnnualTechnologyEmissionByMode_{y,t,e,m,r}$ (E2)
$\forall_{y,t,e,r}$ $AnnualTechnologyEmissionPenaltyByEmission_{y,t,e,r} = AnnualTechnologyEmission_{y,t,e,r} * \mathbf{EmissionsPenalty_{y,e,r}}$ (E3)
$\forall_{y,t,e,r}$ $AnnualTechnologyEmissionsPenalty_{y,t,r} = \sum_m AnnualTechnologyEmissionPenaltyByEmission_{y,t,e,r}$ (E4)
$\forall_{y,e,r}$ $AnnualEmissions_{y,e,r} = \sum_t AnnualTechnologyEmission_{y,t,e,r} + \mathbf{AnnualExogenousEmission_{y,e,r}}$ (E5)
$\forall_{e,r}$ $ModelPeriodEmissions_{e,r} = \sum_y AnnualEmissions_{y,e,r} + \mathbf{ModelPeriodExogenousEmission_{e,r}}$ (E6)
$\forall_{y,e,r}$ $AnnualEmissions_{y,e,r} \leq \mathbf{AnnualEmissionLimit_{y,e,r}}$ (E7)
$\forall_{e,r}$ $ModelPeriodEmissions_{e,r} \leq \mathbf{ModelPeriodEmissionLimit_{e,r}}$ (E8)

Appendix 2. Micro-implementation—the OSeMOSYS 2010 code

The micro-implementation in terms of the full model code is given below. It can effectively be cut and paste into a GNUMathprog model file and run. For more information and a non-pdf version of the code, as well as sample application files, please see: www.osemosys.org. Note also that the "#" symbol use precedes a line of text not used in the model and is included simply for comments. The model code follows:

```
#MODEL DEFINITION#

#SETS#

set YEAR;
set TECHNOLOGY;
set TIMESLICE;
set FUEL;
set EMISSION;
set MODE_OF_OPERATION;
```

```
set REGION;
set BOUNDARY_INSTANCES;
set STORAGE;
```

#PARAMETERS#

#GLOBAL#

```
param StartYear;
param YearSplit{y in YEAR,l in TIMESLICE};
param DiscountRate{t in TECHNOLOGY, r in REGION};
```

#DEMANDS#

```
param SpecifiedAnnualDemand{y in YEAR,f in FUEL, r in REGION};
param SpecifiedDemandProfile{y in YEAR, l in TIMESLICE, f in FUEL, r in REGION};
var RateOfDemand{y in YEAR,l in TIMESLICE, f in FUEL, r in REGION}>= 0;
var Demand{y in YEAR,l in TIMESLICE, f in FUEL, r in REGION}>= 0;
param AccumulatedAnnualDemand{y in YEAR, f in FUEL, r in REGION};
```

#TECHNOLOGY#

#PERFORMANCE#

```
param CapacityToActivityUnit{t in TECHNOLOGY, r in REGION};
param TechWithCapacityNeededToMeetPeakTS{t in TECHNOLOGY, r in REGION};
param CapacityFactor{y in YEAR, t in TECHNOLOGY, r in REGION};
param AvailabilityFactor{y in YEAR, t in TECHNOLOGY, r in REGION};
param OperationalLife{t in TECHNOLOGY, r in REGION};
param ResidualCapacity{y in YEAR, t in TECHNOLOGY, r in REGION};
param SalvageFactor{y in YEAR, t in TECHNOLOGY, r in REGION};
param InputActivityRatio{y in YEAR, t in TECHNOLOGY, f in FUEL, m in MODE_OF_OPERATION, r in REGION};
param OutputActivityRatio{y in YEAR, t in TECHNOLOGY, f in FUEL, m in MODE_OF_OPERATION, r in REGION};
```

#TECHNOLOGY COSTS#

```
param CapitalCost{y in YEAR, t in TECHNOLOGY, r in REGION};
param VariableCost{y in YEAR, t in TECHNOLOGY, m in MODE_OF_OPERATION, r in REGION};
param FixedCost{y in YEAR, t in TECHNOLOGY, r in REGION};
```

#STORAGE PARAMETERS#

```
param StorageInflectionTimes{y in YEAR, l in TIMESLICE, b in BOUNDARY_INSTANCES};
param TechnologyToStorage{t in TECHNOLOGY, m in MODE_OF_OPERATION, s in STORAGE, r in REGION};
param TechnologyFromStorage{t in TECHNOLOGY, m in MODE_OF_OPERATION, s in STORAGE, r in REGION};
param StorageUpperLimit{s in STORAGE, r in REGION};
param StorageLowerLimit{s in STORAGE, r in REGION};
```

#CAPACITY CONSTRAINTS #

```
param TotalAnnualMaxCapacity{y in YEAR, t in TECHNOLOGY, r in REGION};
param TotalAnnualMinCapacity{y in YEAR, t in TECHNOLOGY, r in REGION};
```

#INVESTMENT CONSTRAINTS#

```
param TotalAnnualMaxCapacityInvestment{y in YEAR, t in TECHNOLOGY, r in REGION};
param TotalAnnualMinCapacityInvestment{y in YEAR, t in TECHNOLOGY, r in REGION};
```

#ACTIVITY CONSTRAINTS#

```
param TotalTechnologyAnnualActivityUpperLimit{y in YEAR, t in TECHNOLOGY, r in REGION};
param TotalTechnologyAnnualActivityLowerLimit{y in YEAR, t in TECHNOLOGY, r in REGION};
param TotalTechnologyModelPeriodActivityUpperLimit{t in TECHNOLOGY, r in REGION};
param TotalTechnologyModelPeriodActivityLowerLimit{t in TECHNOLOGY, r in REGION};
```

#RESERVE MARGIN#

```
param ReserveMarginTagTechnology{y in YEAR,t in TECHNOLOGY, r in REGION};
param ReserveMarginTagFuel{y in YEAR,f in FUEL, r in REGION};
param ReserveMargin{y in YEAR, r in REGION};
```

#RE GENERATION TARGET#

```
param RETagTechnology{y in YEAR,t in TECHNOLOGY, r in REGION};
```

```
param RETagFuel{y in YEAR,f in FUEL, r in REGION};
param REMinProductionTarget{y in YEAR, r in REGION};
```

#EMISSIONS & PENALTIES#

```
param EmissionActivityRatio{y in YEAR, t in TECHNOLOGY, e in EMISSION, m in MODE_OF_OPERATION, r in REGION};
param EmissionsPenalty{y in YEAR, e in EMISSION, r in REGION};
param AnnualExogenousEmission{y in YEAR, e in EMISSION, r in REGION};
param AnnualEmissionLimit{y in YEAR, e in EMISSION, r in REGION};
param ModelPeriodExogenousEmission{e in EMISSION, r in REGION};
param ModelPeriodEmissionLimit{e in EMISSION, r in REGION};
```

#MODEL VARIABLES #

#CAPACITY #

```
var NewCapacity{y in YEAR, t in TECHNOLOGY, r in REGION} >= 0;
var AccumulatedNewCapacity{y in YEAR, t in TECHNOLOGY, r in REGION} >= 0;
var TotalCapacityAnnual{y in YEAR, t in TECHNOLOGY, r in REGION}>= 0;
```

#ACTIVITY #

```
var RateOfActivity{y in YEAR, l in TIMESLICE, t in TECHNOLOGY, m in MODE_OF_OPERATION, r in REGION} >= 0;
var RateOfTotalActivity{y in YEAR, l in TIMESLICE, t in TECHNOLOGY, r in REGION} >= 0;
var TotalTechnologyAnnualActivity{y in YEAR, t in TECHNOLOGY, r in REGION} >= 0;
var AverageAnnualTechnologyActivityByMode{y in YEAR, t in TECHNOLOGY,m in MODE_OF_OPERATION,r in REGION}>=0;
var RateOfProductionByTechnologyByMode{y in YEAR, l in TIMESLICE, t in TECHNOLOGY,m in MODE_OF_OPERATION,f in FUEL,r in REGION}>= 0;
var RateOfProductionByTechnology{y in YEAR, l in TIMESLICE, t in TECHNOLOGY,f in FUEL, r in REGION}>= 0;
var ProductionByTechnology{y in YEAR, l in TIMESLICE, t in TECHNOLOGY,f in FUEL, r in REGION}>= 0;
var ProductionByTechnologyAnnual{y in YEAR, t in TECHNOLOGY, f in FUEL, r in REGION}>= 0;
var RateOfProduction{y in YEAR, l in TIMESLICE, f in FUEL, r in REGION} >= 0;
var Production{y in YEAR, l in TIMESLICE, f in FUEL, r in REGION} >= 0;
var RateOfUseByTechnologyByMode{y in YEAR, l in TIMESLICE, t in TECHNOLOGY,m in MODE_OF_OPERATION,f in FUEL,r in REGION}>= 0;
var RateOfUseByTechnology{y in YEAR, l in TIMESLICE, t in TECHNOLOGY, f in FUEL, r in REGION} >= 0;
var UseByTechnologyAnnual{y in YEAR, t in TECHNOLOGY,f in FUEL, r in REGION}>= 0;
var RateOfUse{y in YEAR, l in TIMESLICE, f in FUEL, r in REGION}>= 0;
var UseByTechnology{y in YEAR, l in TIMESLICE, t in TECHNOLOGY,f in FUEL, r in REGION}>= 0;
var Use{y in YEAR, l in TIMESLICE, f in FUEL, r in REGION}>= 0;
#
var ProductionAnnual{y in YEAR, f in FUEL, r in REGION}>= 0;
var UseAnnual{y in YEAR, f in FUEL, r in REGION}>= 0;
```

#COSTING#

```
var CapitalInvestment{y in YEAR, t in TECHNOLOGY, r in REGION}>= 0;
var DiscountedCapitalInvestment{y in YEAR, t in TECHNOLOGY, r in REGION}>= 0;
var SalvageValue{y in YEAR, t in TECHNOLOGY, r in REGION}>= 0;
var DiscountedSalvageValue{y in YEAR, t in TECHNOLOGY, r in REGION}>= 0;
var OperatingCost{y in YEAR, t in TECHNOLOGY, r in REGION}>= 0;
var DiscountedOperatingCost{y in YEAR, t in TECHNOLOGY, r in REGION}>= 0;
var AnnualVariableOperatingCost{y in YEAR,t in TECHNOLOGY, r in REGION}>= 0;
var AnnualFixedOperatingCost{y in YEAR,t in TECHNOLOGY, r in REGION}>= 0;
var VariableOperatingCost{y in YEAR, l in TIMESLICE, t in TECHNOLOGY, r in REGION}>= 0;
var TotalDiscountedCost{y in YEAR, t in TECHNOLOGY, r in REGION}>= 0;
var ModelPeriodCostByRegion {r in REGION} >= 0;
```

STORAGE

```
var NetStorageCharge{s in STORAGE, y in YEAR, l in TIMESLICE, r in REGION};
var StorageLevel{s in STORAGE, b in BOUNDARY_INSTANCES, r in REGION};
var StorageCharge{s in STORAGE, y in YEAR, l in TIMESLICE, r in REGION};
var StorageDischarge{s in STORAGE, y in YEAR, l in TIMESLICE, r in REGION};
```

RESERVE MARGIN

```
var TotalCapacityInReserveMargin{y in YEAR, r in REGION}>= 0;
var DemandNeedingReserveMargin{y in YEAR,l in TIMESLICE, r in REGION}>= 0;
```

RENEWABLE ENERGY PRODUCTION TARGET

```
var TotalGenerationByRETechnologies{y in YEAR, r in REGION};
var TotalREProductionAnnual{y in YEAR, r in REGION};
var RETotalDemandOfTargetFuelAnnual{y in YEAR  r in REGION};
```

```
var TotalTechnologyModelPeriodActivity{t in TECHNOLOGY, r in REGION};
```

EMISSIONS

```
var AnnualTechnologyEmissionByMode{y in YEAR, t in TECHNOLOGY, e in EMISSION, m in MODE_OF_OPERATION, r in REGION}>= 0;
var AnnualTechnologyEmission{y in YEAR, t in TECHNOLOGY, e in EMISSION, r in REGION}>= 0;
var AnnualTechnologyEmissionPenaltyByEmission{y in YEAR, t in TECHNOLOGY, e in EMISSION, r in REGION}>= 0;
var AnnualTechnologyEmissionsPenalty{y in YEAR, t in TECHNOLOGY, r in REGION}>= 0;
var AnnualEmissions{y in YEAR, e in EMISSION, r in REGION}>= 0;
var EmissionsProduction{y in YEAR, t in TECHNOLOGY, e in EMISSION, m in MODE_OF_OPERATION, r in REGION};
var ModelPeriodEmissions{e in EMISSION, r in REGION}>= 0;
```

OBJECTIVE

```
minimize OBJ_TotalNPVCost: sum{y in YEAR, t in TECHNOLOGY, r in REGION} TotalDiscountedCost[y,t,r];
```

CONSTRAINTS AND EQUATIONS

TOTAL DISCOUNTED COSTS

```
s.t. TDC1_TotalDiscountedCostByTechnology{y in YEAR, t in TECHNOLOGY, r in REGION}:
DiscountedOperatingCost[y,t,r]+DiscountedCapitalInvestment[y,t,r] + AnnualTechnologyEmissionsPenalty[y,t,r]- DiscountedSalvageValue[y,t,r] =
TotalDiscountedCost[y,t,r];
```

OPERATING COSTS

```
s.t. OC1_OperatingCostsVariable{y in YEAR,l in TIMESLICE, t in TECHNOLOGY, r in REGION}: sum{m in MODE_OF_OPERATION}
    RateOfActivity[y,l,t,m,r]*VariableCost[y,t,m,r] = VariableOperatingCost[y,l,t,r];
s.t. OC2_OperatingCostsVariableAnnual{y in YEAR,t in TECHNOLOGY, r in REGION}: sum {l in TIMESLICE} VariableOperatingCost[y,l,t,r] =
    AnnualVariableOperatingCost[y,t,r];
s.t. OC3_OperatingCostsFixedAnnual{y in YEAR,t in TECHNOLOGY, r in REGION}: TotalCapacityAnnual[y,t,r]*FixedCost[y,t,r] =
    AnnualFixedOperatingCost[y,t,r];
s.t. OC4_OperatingCostsTotalAnnual{y in YEAR,t in TECHNOLOGY,r in REGION}: AnnualFixedOperatingCost[y,t,r]+AnnualVariableOperatingCost[y,t,r] =
    OperatingCost[y,t,r];
s.t. OC5_DiscountedOperatingCostsTotalAnnual{y in YEAR, t in TECHNOLOGY, r in REGION}: OperatingCost[y,t,r]/((1+DiscountRate[t,r])^(y-
    StartYear+0.5)) = DiscountedOperatingCost[y,t,r];
```

CAPTIAL COSTS

```
s.t. CC1_UndiscountedCapitalInvestment{y in YEAR, t in TECHNOLOGY, r in REGION}: CapitalCost[y,t,r] * NewCapacity[y,t,r] = CapitalInvestment[y,t,r];
s.t. CC2_DiscountingCapitalInvestmenta{y in YEAR, t in TECHNOLOGY, r in REGION}: CapitalInvestment[y,t,r]/((1+DiscountRate[t,r])^(y-StartYear)) =
    DiscountedCapitalInvestment[y,t,r];
```

SALVAGE VALUE

```
s.t. SV1_SalvageValueAtEndOfPeriod2{y in YEAR, t in TECHNOLOGY, r in REGION: (y + OperationalLife[t,r]) < (StartYear + card(YEAR))}: SalvageValue[y,t,r]
    = 0;
s.t. SV2_SalvageValueAtEndOfPeriod1{y in YEAR, t in TECHNOLOGY, r in REGION: (y + OperationalLife[t,r]) >= (StartYear + card(YEAR))}:
    SalvageValue[y,t,r] = NewCapacity[y,t,r]* CapitalCost[y,t,r]* CapitalCost[y,t,r]*(1-((((1+DiscountRate[t,r])^(StartYear+card(YEAR) - y))-
    1)/((1+DiscountRate[t,r])^OperationalLife[t,r]-1)));
s.t. SV3_SalvageValueDiscountedToStartYear{y in YEAR, t in TECHNOLOGY, r in REGION}: DiscountedSalvageValue[y,t,r] =
    SalvageValue[y,t,r]/((1+DiscountRate[t,r])^(1+card(YEAR)));
```

DEMAND

```
s.t. EQ_SpecifiedDemand1{y in YEAR,l in TIMESLICE, f in FUEL, r in REGION:SpecifiedAnnualDemand[y,f,r]<>0}:
    SpecifiedAnnualDemand[y,f,r]*SpecifiedDemandProfile[y,l,f,r] / YearSplit[y,l]=RateOfDemand[y,l,f,r];
```

STORAGE

```
s.t. S1_StorageCharge{s in STORAGE, y in YEAR, l in TIMESLICE, r in REGION}: sum{t in TECHNOLOGY, m in MODE_OF_OPERATION}
    RateOfActivity[y,l,t,m,r] * TechnologyToStorage[t,m,s,r] * YearSplit[y,l] = StorageCharge[s,y,l,r];
s.t. S2_StorageDischarge{s in STORAGE, y in YEAR, l in TIMESLICE, r in REGION}: sum{t in TECHNOLOGY, m in MODE_OF_OPERATION}
    RateOfActivity[y,l,t,m,r] * TechnologyFromStorage[t,m,s,r] * YearSplit[y,l] = StorageDischarge[s,y,l,r];
s.t. S3_NetStorageCharge{s in STORAGE, y in YEAR, l in TIMESLICE, r in REGION}: NetStorageCharge[s,y,l,r] = StorageCharge[s,y,l,r] -
    StorageDischarge[s,y,l,r];
s.t. S4_StorageLevelAtInflection{b in BOUNDARY_INSTANCES,s in STORAGE,r in REGION}: sum{l in TIMESLICE, y in YEAR}
    NetStorageCharge[s,y,l,r]/YearSplit[y,l] *StorageInflectionTimes[y,l,b] = StorageLevel[s,b,r];
s.t. S5_StorageLowerLimit{b in BOUNDARY_INSTANCES, s in STORAGE,r in REGION}: StorageLevel[s,b,r] >= StorageLowerLimit[s,r];
s.t. S6_StorageUpperLimit{b in BOUNDARY_INSTANCES, s in STORAGE,r in REGION}: StorageLevel[s,b,r] <= StorageUpperLimit[s,r];
```

CAPACITY ADEQUACY “A”

```
s.t. CAa1_TotalNewCapacity{y in YEAR, t in TECHNOLOGY, r in REGION}:AccumulatedNewCapacity[y,t,r] = sum{yy in YEAR: y-yy < OperationalLife[t,r] && y-yy>=0} NewCapacity[yy,t,r];
s.t. CAa2_TotalAnnualCapacity{y in YEAR, t in TECHNOLOGY, r in REGION}: AccumulatedNewCapacity[y,t,r]+ ResidualCapacity[y,t,r] = TotalCapacityAnnual[y,t,r];
s.t. CAa3_TotalActivityOfEachTechnology{y in YEAR, t in TECHNOLOGY, l in TIMESLICE,r in REGION}: sum{m in MODE_OF_OPERATION} RateOfActivity[y,l,t,m,r] = RateOfTotalActivity[y,l,t,r];
s.t. CAa4_Constraint_Capacity{y in YEAR, l in TIMESLICE, t in TECHNOLOGY, r in REGION: TechWithCapacityNeededToMeetPeakTS[t,r]<>0}: RateOfTotalActivity[y,l,t,r] <= TotalCapacityAnnual[y,t,r] * CapacityFactor[y,t,r]*CapacityToActivityUnit[t,r];
# Note that the PlannedMaintenance equation below ensures that all other technologies have a capacity great enough to at least meet the annual average.

# CAPACITY ADEQUACY "B" #

s.t. CAb1_PlannedMaintenance{y in YEAR, t in TECHNOLOGY, r in REGION}: sum{l in TIMESLICE} RateOfTotalActivity[y,l,t,r]*YearSplit[y,l] <= TotalCapacityAnnual[y,t,r]*CapacityFactor[y,t,r]* AvailabilityFactor[y,t,r]*CapacityToActivityUnit[t,r];

# ENERGY BALANCE A #

s.t. EBa1_RateOfFuelProduction1{y in YEAR, l in TIMESLICE, f in FUEL, t in TECHNOLOGY, m in MODE_OF_OPERATION, r in REGION}: RateOfActivity[y,l,t,m,r]*OutputActivityRatio[y,t,f,m,r]  = RateOfProductionByTechnologyByMode[y,l,t,m,f,r];
s.t. EBa2_RateOfFuelProduction2{y in YEAR, l in TIMESLICE, f in FUEL, t in TECHNOLOGY, r in REGION}: sum{m in MODE_OF_OPERATION} RateOfProductionByTechnologyByMode[y,l,t,m,f,r] = RateOfProductionByTechnology[y,l,t,f,r] ;
s.t. EBa3_RateOfFuelProduction3{y in YEAR, l in TIMESLICE, f in FUEL, r in REGION}: sum{t in TECHNOLOGY} RateOfProductionByTechnology[y,l,t,f,r]  = RateOfProduction[y,l,f,r];
s.t. EBa4_RateOfFuelUse1{y in YEAR, l in TIMESLICE, f in FUEL, t in TECHNOLOGY, m in MODE_OF_OPERATION, r in REGION}: RateOfActivity[y,l,t,m,r]*InputActivityRatio[y,t,f,m,r]  = RateOfUseByTechnologyByMode[y,l,t,m,f,r];
s.t. EBa5_RateOfFuelUse2{y in YEAR, l in TIMESLICE, f in FUEL, t in TECHNOLOGY, r in REGION}: sum{m in MODE_OF_OPERATION} RateOfUseByTechnologyByMode[y,l,t,m,f,r] = RateOfUseByTechnology[y,l,t,f,r];
s.t. EBa6_RateOfFuelUse3{y in YEAR, l in TIMESLICE, f in FUEL, r in REGION}: sum{t in TECHNOLOGY} RateOfUseByTechnology[y,l,t,f,r]  = RateOfUse[y,l,f,r];
s.t. EBa7_EnergyBalanceEachTS1{y in YEAR,l in TIMESLICE, f in FUEL, r in REGION}: RateOfProduction[y,l,f,r]*YearSplit[y,l] = Production[y,l,f,r];
s.t. EBa8_EnergyBalanceEachTS2{y in YEAR,l in TIMESLICE, f in FUEL, r in REGION}: RateOfUse[y,l,f,r]*YearSplit[y,l] = Use[y,l,f,r];
s.t. EBa9_EnergyBalanceEachTS3{y in YEAR,l in TIMESLICE, f in FUEL, r in REGION}: RateOfDemand[y,l,f,r]*YearSplit[y,l] = Demand[y,l,f,r];
s.t. EBa10_EnergyBalanceEachTS4{y in YEAR,l in TIMESLICE, f in FUEL, r in REGION}: Production[y,l,f,r] >= Demand[y,l,f,r] + Use[y,l,f,r];

# ENERGY BALANCE B #

s.t. EBb1_EnergyBalanceEachYear1{y in YEAR, f in FUEL, r in REGION}: sum{l in TIMESLICE} Production[y,l,f,r] = ProductionAnnual[y,f,r];
s.t. EBb2_EnergyBalanceEachYear2{y in YEAR, f in FUEL, r in REGION}: sum{l in TIMESLICE} Use[y,l,f,r] = UseAnnual[y,f,r];
s.t. EBb3_EnergyBalanceEachYear3{y in YEAR, f in FUEL, r in REGION}: ProductionAnnual[y,f,r] >= UseAnnual[y,f,r] + AccumulatedAnnualDemand[y,f,r];

# ACCOUNTING TECHNOLOGY PRODUCTION/USE #

s.t. Acc1_FuelProductionByTechnology{y in YEAR, l in TIMESLICE, t in TECHNOLOGY, f in FUEL, r in REGION}: RateOfProductionByTechnology[y,l,t,f,r] * YearSplit[y,l] = ProductionByTechnology[y,l,t,f,r];
s.t. Acc2_FuelUseByTechnology{y in YEAR, l in TIMESLICE, t in TECHNOLOGY, f in FUEL, r in REGION}: RateOfUseByTechnology[y,l,t,f,r] * YearSplit[y,l] = UseByTechnology[y,l,t,f,r];
s.t. Acc3_AverageAnnualRateOfActivity{y in YEAR,l in TIMESLICE, t in TECHNOLOGY, m in MODE_OF_OPERATION, r in REGION}: RateOfActivity[y,l,t,m,r]*YearSplit[y,l] = AverageAnnualTechnologyActivityByMode[y,t,m,r];
s.t. Acc3_ModelPeriodCostByRegion{r in REGION}:sum{y in YEAR, t in TECHNOLOGY}TotalDiscountedCost[y,t,r]=ModelPeriodCostByRegion[r];

#ACTIVITY, CAPACITY, RESERVE MARGIN AND RENEWABLE PRODUCTION CONSTRAINTS#

# TOTAL CAPACITY CONSTRAINTS  #

s.t. TCC1_TotalAnnualMaxCapacityConstraint{y in YEAR, t in TECHNOLOGY,r in REGION: TotalAnnualMaxCapacity[y,t,r]<9999 }: TotalCapacityAnnual[y,t,r] <= TotalAnnualMaxCapacity[y,t,r];
s.t. TCC2_TotalAnnualMinCapacityConstraint{y in YEAR, t in TECHNOLOGY,r in REGION: TotalAnnualMinCapacity[y,t,r]>0}: TotalCapacityAnnual[y,t,r] >= TotalAnnualMinCapacity[y,t,r];

# NEW CAPACITY CONSTRAINTS #

s.t. NCC1_TotalAnnualMaxNewCapacityConstraint{y in YEAR, t in TECHNOLOGY, r in REGION: TotalAnnualMaxCapacityInvestment[y,t,r]<9999}: NewCapacity[y,t,r] <= TotalAnnualMaxCapacityInvestment[y,t,r];
s.t. NCC2_TotalAnnualMinNewCapacityConstraint{y in YEAR, t in TECHNOLOGY, r in REGION: TotalAnnualMinCapacityInvestment[y,t,r]>0}: NewCapacity[y,t,r] >= TotalAnnualMinCapacityInvestment[y,t,r];

# ANNUAL ACTIVITY CONSTRAINTS  #

s.t. AAC1_TotalAnnualTechnologyActivity{y in YEAR, t in TECHNOLOGY, r in REGION}: sum{l in TIMESLICE} RateOfTotalActivity[y,l,t,r]*YearSplit[y,l] = TotalTechnologyAnnualActivity[y,t,r];
s.t. AAC2_TotalAnnualTechnologyActivityUpperLimit{y in YEAR, t in TECHNOLOGY, r in REGION:TotalTechnologyAnnualActivityUpperLimit[y,t,r]<9999}: TotalTechnologyAnnualActivity[y,t,r] <= TotalTechnologyAnnualActivityUpperLimit[y,t,r] ;
s.t. AAC3_TotalAnnualTechnologyActivityLowerLimit{y in YEAR, t in TECHNOLOGY, r in REGION: TotalTechnologyAnnualActivityLowerLimit[y,t,r]>0}: TotalTechnologyAnnualActivity[y,t,r] >= TotalTechnologyAnnualActivityLowerLimit[y,t,r] ;
```

```
# TOTAL ACTIVITY CONSTRAINTS #

s.t. TAC1_TotalModelHorizenTechnologyActivity{t in TECHNOLOGY, r in REGION}: sum{y in YEAR} TotalTechnologyAnnualActivity[y,t,r] =
    TotalTechnologyModelPeriodActivity[t,r];
s.t. TAC2_TotalModelHorizenTechnologyActivityUpperLimit{y in YEAR, t in TECHNOLOGY, r in
    REGION:TotalTechnologyModelPeriodActivityUpperLimit[t,r]<9999}: TotalTechnologyModelPeriodActivity[t,r] <=
    TotalTechnologyModelPeriodActivityUpperLimit[t,r] ;
s.t. TAC3_TotalModelHorizenTechnologyActivityLowerLimit{y in YEAR, t in TECHNOLOGY, r in REGION:
    TotalTechnologyModelPeriodActivityLowerLimit[t,r]>0}: TotalTechnologyModelPeriodActivity[t,r] >=
    TotalTechnologyModelPeriodActivityLowerLimit[t,r] ;

# RESERVE MARGIN CONSTRAINT #

s.t. RM1_ReserveMargin_TechologiesIncluded_In_Activity_Units{y in YEAR, l in TIMESLICE, r in REGION}: sum {t in TECHNOLOGY}
    TotalCapacityAnnual[y,t,r] *ReserveMarginTagTechnology[y,t,r] * CapacityToActivityUnit[t,r]      =      TotalCapacityInReserveMargin[y,r];
s.t. RM2_ReserveMargin_FuelsIncluded{y in YEAR, l in TIMESLICE, r in REGION}: sum {f in FUEL} RateOfDemand[y,l,f,r] * ReserveMarginTagFuel[y,f,r] =
    DemandNeedingReserveMargin[y,l,r];
s.t. RM3_ReserveMargin_Constraint{y in YEAR, l in TIMESLICE, r in REGION}: DemandNeedingReserveMargin[y,l,r] * ReserveMargin[y,r]<=
    TotalCapacityInReserveMargin[y,r];

# RE PRODUCTION TARGET #

s.t. RE1_FuelProductionByTechnologyAnnual{y in YEAR, t in TECHNOLOGY, f in FUEL, r in REGION}: sum{l in TIMESLICE} ProductionByTechnology[y,l,t,f,r]
    = ProductionByTechnologyAnnual[y,t,f,r];
s.t. RE2_TechIncluded{y in YEAR, r in REGION}: sum{t in TECHNOLOGY, f in FUEL} ProductionByTechnologyAnnual[y,t,f,r]*RETagTechnology[y,t,r] =
    TotalREProductionAnnual[y,r];
s.t. RE3_FuelIncluded{y in YEAR, r in REGION}: sum{l in TIMESLICE, f in FUEL} RateOfDemand[y,l,f,r]*YearSplit[y,l]*RETagFuel[y,f,r] =
    RETotalDemandOfTargetFuelAnnual[y,r];
s.t. RE4_EnergyConstraint{y in YEAR, r in REGION}:REMinProductionTarget[y,r]*RETotalDemandOfTargetFuelAnnual[y,r] <=
    TotalREProductionAnnual[y,r];
s.t. RE5_FuelUseByTechnologyAnnual{y in YEAR, t in TECHNOLOGY, f in FUEL, r in REGION}: sum{l in TIMESLICE}
    RateOfUseByTechnology[y,l,t,f,r]*YearSplit[y,l] = UseByTechnologyAnnual[y,t,f,r];

# EMISSIONS ACCOUNTING#

s.t. E1_AnnualEmissionProductionByMode{y in YEAR, t in TECHNOLOGY, e in EMISSION, m in MODE_OF_OPERATION, r in
    REGION:EmissionActivityRatio[y,t,e,m,r]<>0}: sum{l in TIMESLICE}
    EmissionActivityRatio[y,t,e,m,r]*AverageAnnualTechnologyActivityByMode[y,t,m,r]=AnnualTechnologyEmissionByMode[y,t,e,m,r];
s.t. E2_AnnualEmissionProduction{y in YEAR, t in TECHNOLOGY, e in EMISSION, r in REGION}: sum{m in MODE_OF_OPERATION}
    AnnualTechnologyEmissionByMode[y,t,e,m,r] = AnnualTechnologyEmission[y,t,e,r];
s.t. E3_EmissionsPenaltyByTechAndEmission{y in YEAR, t in TECHNOLOGY, e in EMISSION, r in REGION}:
    AnnualTechnologyEmission[y,t,e,r]*EmissionsPenalty[y,e,r] = AnnualTechnologyEmissionPenaltyByEmission[y,t,e,r];
s.t. E4_EmissionsPenaltyByTechnology{y in YEAR, t in TECHNOLOGY, r in REGION}: sum{e in EMISSION}
    AnnualTechnologyEmissionPenaltyByEmission[y,t,e,r] = AnnualTechnologyEmissionsPenalty[y,t,r];
s.t. E5_EmissionsAccounting1{y in YEAR, e in EMISSION, r in REGION}: sum{t in TECHNOLOGY} AnnualTechnologyEmission[y,t,e,r] =
    AnnualEmissions[y,e,r] - AnnualExogenousEmission[y,e,r];
s.t. E6_EmissionsAccounting2{e in EMISSION, r in REGION}: sum{y in YEAR} AnnualEmissions[y,e,r] = ModelPeriodEmissions[e,r] -
    ModelPeriodExogenousEmission[e,r];
s.t. E7_AnnualEmissionsLimit{y in YEAR, e in EMISSION, r in REGION}: AnnualEmissions[y,e,r] <= AnnualEmissionLimit[y,e,r];
s.t. E8_ModelPeriodEmissionsLimit{e in EMISSION, r in REGION}:  ModelPeriodEmissions [e,r] <= ModelPeriodEmissionLimit[e,r];

#
###########################################################################################
#
solve;
#
###########################################################################################
end;
```

References

AMPL, 2010. AMPL Modeling Language for Mathematical Programming. 〈www.ampl.com〉 (accessed September).

Apache, 2004. Apache License, Version 2.0. 〈http://www.apache.org/licenses/LICENSE-2.0.html〉.

Bazillian, M., Sagar, A., Detchon, R., Yumkella, K., 2010. Moor heat and light. Energy Policy 38 (10), 5409–5412.

Bella, M., Doernberga, A., Hermeleea, A., Hoffman, K., 1979. Impacts of new energy technologies as measured against reference energy systems. Energy 4 (5), 891–909.

COMMEND 2010. 〈www.energycommunity.org〉 (accessed September).

DeCarolis, J., Hunter, K., Sreepathi, S., 2010. The TEMOA Project: Tools for Energy Model Optimization and Analysis. International Energy Workshop 2010, Stockholm, Sweden, June 21–23, 2010.

Enerdata, 2010. POLES: Prospective Outpook on Long Term Energy Systems. 〈http://www.enerdata.net/enerdatauk/tools/Model_POLES.html〉 (accessed September).

ETSAP (Energy Technology Systems Analysis Program), 2010. Software and Tools. 〈http://www.etsap.org/Tools.asp〉 (accessed September).

GAMS, 2010. GAMS General Algebraic Modeling System. 〈www.gams.com〉 (accessed September).

GNU Mathprog, 2009. 〈http://lpsolve.sourceforge.net/5.5/MathProg.htm〉 (accessed February).

Heaps, C., Howells, M., Clark, V., 2011. Making Energy System Optimization Methodologies Accessible and Affordable: Incorporating the Open Source Energy Modeling System (OSeMOSYS) into the Long Range Eergy Alternatives Planning System (LEAP). 〈http://osmosys.yolasite.com/leap-osemosys.php〉. KTH (2011), 2011 Steering Committee Meeting, 〈http://osmosys.yolasite.com/2011-osemoysy-steering-committee-meeting.php〉, (accessed March).

Hermann, S., Howells, M., Rogner, H., Bazilian, M., Welsch, M., 2011. The CLEW Model — Developing an integrated tool for modeling the interrelated effects between Climate, Land use, Energy, and Water (CLEW). 〈http://osemosysmain.yolasite.com/resources/Multi%20Resource%20CLEW%20modelling%20in%20OSeMOSYS.pdf〉.

Howells, M., Alfstad, T., Victor, D., Goldstein, G., Remme, U., 2005. A model of household energy services in a low-income rural African village. Energy Policy 33 (14), 1833–1851.

Howells, M., Rogner, H., Jalal, I., Isshiki, 2008. Development of an 'Open-source' Energy Planning Methodology, International Energy Workshop, Paris, June. 〈http://www.iea.org/Textbase/work/2008/iew/Wednesday/Howells.pdf〉.

Howells, M., Rogner, H., Strachan, N., Heaps, C., Huntington, H., Kypreos, S., Hughes, A., Silveira, S., DeCarolis, J., Bazillian, M. and Rhoerl, A., 2010. OSeMOSYS: The Open Source Energy Modelling System. International Energy Workshop. Stockholm.

Howells, M., Weisser, D., Rogner, H., 2011. Modelling Energy Supply Security in the Power Sector: Notes on Generic Policy Options. 〈http://osmosys.yolasite.com/modelling-energy-security.php〉.

Huntington, H., Weyant, J., Sweeney, J., 1982. Modeling for insights, not numbers: the experiences of the energy modeling forum. Omega 10 (5), 449–462.

IAEA (International Atomic Energy Agency), 2010. PESS Energy Models. 〈http://www.iaea.org/OurWork/ST/NE/Pess/PESSenergymodels.shtml〉, (accessed September).

IIASA (International Institute of Applied Systems Analysis), 2001. Model Message—Command Line User Manual. 〈http://www.iiasa.ac.at/Research/ECS/docs/MESSAGE_man018.pdf〉.

KTH (Royal Swedish Institute of Technology), 2009. Energy Scenarios. Ph.D. Course.

KTH (Royal Swedish Institute of Technology), 2010. OSeMOSYS Workshop. 〈http://www.kth.se/en/itm/konferenser/genomforda-konferenser/iew2010/iew2010news/osemosys-workshop-june-22nd-1.56490〉.

KTH, 2011. 2011 Steering Committee Meeting. 〈http://osmosys.yolasite.com/2011-osemoysy-steering-committee-meeting.php〉 (accessed March).

Loulou, R., Goldstein, G., Knoble, K., 2004. Documentation for the MARKAL Family of Models. ETSAP. 〈http://www.etsap.org/MrklDoc-I_StdMARKAL.pdf〉.

Makhorin, A., 2003. GNU Linear Programming Kit Modeling Language GNU MathProg. 〈http://www.lix.polytechnique.fr/~liberti/teaching/amaldi/fro/MPL_language.pdf〉.

Maximal, 2010. GLPK for MPL. 〈http://www.maximalsoftware.com/solvers/glpk.html〉 (accessed September).

McConnell, S., 2004. Code Complete 2nd ed. Microsoft Press, ISBN: 978-0735619678.

Munson, R., 2004. Improving Prediction of Energy Futures. Issues in Science and Technology. 〈http://www.issues.org/20.3/p_munson.html〉, Spring.

Murtagh, B., 1981. Advanced Linear Programming: Computation and Practice. McGraw-Hill.

NTUA (National Technical University of Athens), 2010. The PRIMES Energy Systems Model. 〈http://www.e3mlab.ntua.gr/manuals/PRIMsd.pdf〉 (accessed September).

OSeMOSYS, 2009. The Open Source Energy Modeling System. OSeMOSYS (accessed February).

Strachan, N., Pye S., S., Kannan R, R., 2009. The iterative contribution and relevance of modeling to UK Energy Policy. Energy Policy 37 (3), 850–860.

Strubegger, M., 2010. Current developer of MESSAGE. Personal Communication.

TEMOA [Tools for Energy Model Optimization and Analysis], 2011. 〈http://www.temoaproject.org/〉 (accessed July).

UK ERC (United Kingdom Energy Research Centre), 2010. OSeMOSYS Workshop 22–24 February, Oxford. 〈http://www.ukerc.ac.uk/support/tiki-index.php?page_ref_id=2573〉.

USDOE (United States Department of Energy), 2003. Model Documentation Report: System for the Analysis of Global Energy Markets (SAGE). 〈http://tonto.eia.doe.gov/ftproot/modeldoc/m072(2003)1.pdf〉 (*Note*: the actual GAMS Code for this MARKAL based system can be requested here: 〈http://tonto.eia.doe.gov/aeo/modelrequest/modreq.asp〉.

Van der Voort, E., 1982. The EFOM 12C energy supply model within the EC modeling system. Omega 10 (5), 507–5231982, 507–523.

Welsch, M., Howells, M., Bazilian, M., Hermann, S., Rogner, H., 2011. Modelling Elements of Smart Grids for sub-Saharan African Countries, 〈http://osmosys.yolasite.com/smart-grids.php〉.