Creating Suites of Models with System Entity Structure: Global Warming Example

Bernard P. Zeigler, Chungman Seo, Robert Coop, and Doohwan Kim

RTSync Corp.
530 Bartow Drive Suite A
Sierra Vista, AZ, 85635, USA
zeigler, cseo, robert.coop, dhkim@rtsync.com

Keywords: System Entity Structure, Suite of Models, component-based modeling, Systems of systems

Abstract

We describe how to develop a suite of models in the MS4 Modeling Environment. The approach employs the operation of merging of System Entity Structures supported by the environment. After construction, the suite of models can be hosted on Model Store, the cloud-based repository of models provided by MS4 systems as a basis for further collaborative model development. A suite of models, relating to Global Warming is used as an example. We discuss enhanced browsing as necessary to enable a developer to work effectively in a marketplace environment.

1. INTRODUCTION

As has been described in a number of publications (see e.g., [1-5]) the System Entity Structure (SES) supports development, pruning, and generation of a family of simulation models, In this paper, we introduce an expanded concept involving multiple families supported by MS4 Me [6], the Modeling Environment developed by MS4Systems.com . Figure 1 contrasts a suite of models with a single family of models. In Figure 1a) an SES implements a single family of models while in Figure 1b) a set of non-overlapping SESs represents a set of families of unrelated models. Finally Figure 1c) depicts a set of intersecting SESs representing a suite of related families of models. In such a suite, there are SESs that are "components of other SESs. This use of the term "components" transfers the "component of" concept from its use in component-based model construction [7] to the domain of SES construction. Thus, an SES is a component of another SES in the sense that the models the first SES generates are components of models generated by the second SES. The operation of composing DEVS models to create a coupled model [8] is mirrored by the merging operation for composing SESs [5]. As will be explained, merging generalizes the DEVS construction process in which individual models can be developed, tested and then composed to create hierarchical models in stage-wise fashion. This is to say, merging supports hierarchical composition in which families of models are generated and tested via pruning and transforming their SESs in bottom-up manner.

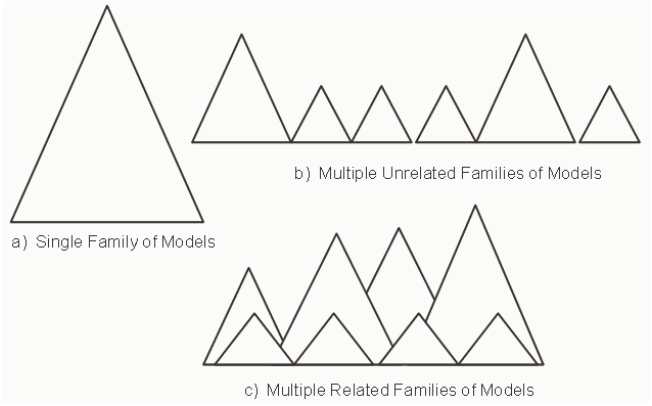


Figure 1 Suites of Models supported by the SES

2. GLOBAL WARMING SUITE OF MODELS

Global warming provides a domain where suites of models are related by common carbon-driven warming processes potentially causing a variety of climate-variation effects. The following is an example of such families of models and the questions they address:

- **Greenhouse effect Model Family**: What is the greenhouse effect that causes warming of the earth?
- **PolarIce Melting Model Family**: How does global warming cause increased melting of the Polar Ice Cap?
- **PermaFrost Melting Model Family:** How does the warming of the permafrost contribute to ever increasing global warming?
- **SeaLevel Rising Model Family**: How does the global warming contribute to the rising of the sea level?

- **Storm Intensity Increasing Model Family**: How does the global warming contribute to the increasing intensity of storms?
- **Flood Increasing Model Family**: How does the global warming contribute to the increasing incidence of floods?
- **Drought Increasing Model Family**: How does the global warming contribute to the increasing incidence of droughts?

Appendix 1 contains more description of the models in this collection. Each model family is a set of models pruned and generated from a corresponding SES. As illustrated in Figure 2, these SESs form a set that is related by a composition relation. Here an arrow indicates composition, i.e., an SES is composed of the SESs and atomic entities sending arrows to it. For example, the GreenHouseEffect SES is composed only of atomic entities while it is a component in many other SESs, reflecting the case that greenhouse gases are a cause of the related climate change phenomena.

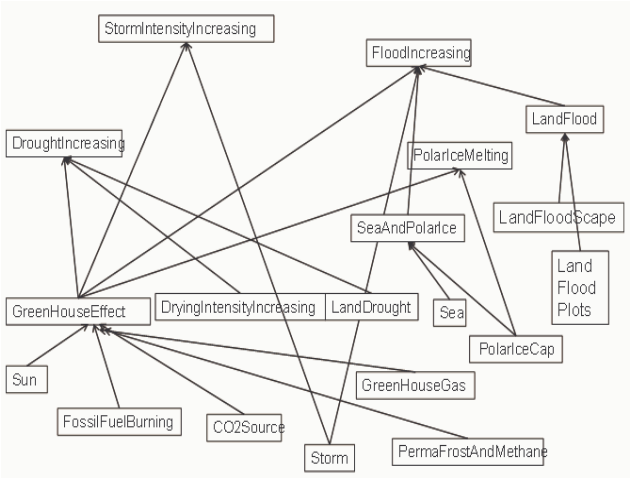


Figure 2 Suite of Models for Global Warming

3. UNDERSTANDING THE SYSTEM ENTITY STRUCTURE

To explain how to develop model families we start with manual creation of DEVS coupled models and evolve to the next level where the SES automates most of this work. To start, consider the Flood Increasing model family, an instance of which is illustrated in Figure 3. Here the components are the model families for Sea and Polar Ice ,Record and Graph Flooding, Greenhouse Effect , Storm and Land Flood The couplings are depicted by arrows from output ports to input ports. For example, there is a coupling from the output port, outRain of Storm to the input port, inRain of LandPlot.

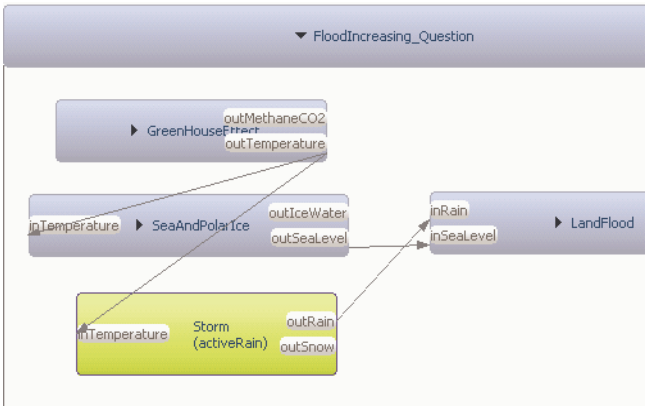


Figure 3 Flood increasing model

A common way of creating a coupled model requires defining a class for it with a constructor that adds its components and their couplings. For example,

public FloodIncreasing(){

//create instances of component classes and add them to the model

```
SeaAndPolarIce SeaAndPolarIce = new
SeaAndPolarIce();
addChildModel(SeaAndPolarIce);

RecordAndGraphFlooding
RecordAndGraphFlooding = new
RecordAndGraphFlooding();
addChildModel(RecordAndGraphFlooding);

GreenHouseEffect GreenHouseEffect = new
GreenHouseEffect();
addChildModel(GreenHouseEffect);

Storm Storm = new Storm();
addChildModel(Storm);

LandFlood LandFlood = new LandFlood();
addChildModel(LandFlood);
```

// add the couplings

```
addCoupling(LandFlood.outFloodLevel,RecordAn
dGraphFlooding.inFloodLevel);
addCoupling(Storm.outRain,LandFlood.inRain);
addCoupling(GreenHouseEffect.outTemperature,
Storm.inTemperature);
addCoupling(GreenHouseEffect.outTemperature,
SeaAndPolarIce.inTemperature);
addCoupling(SeaAndPolarIce.outSeaLevel,LandF
lood.inSeaLeve);
}
```

The SES generates class definition containing these kinds of imperative commands from declarative statements in the form of constrained natural language sentences. For

example, to generate the same model you can write the following SES specification:

//describe the components

```
From the FloodIncreasingsys perspective,
FloodIncreasing is made of GreenHouseEffect,
Storm, SeaAndPolarIce, LandFlood, and
RecordAndGraphFlooding!
```

//describe the couplings

```
From the FloodIncreasingsys perspective,
GreenHouseEffect sends Temperature to Storm!
From the FloodIncreasingsys perspective,
Storm sends Rain to LandFlood!
From the FloodIncreasingsys perspective,
GreenHouseEffect sends Temperature to
SeaAndPolarIce!
From the FloodIncreasingsys perspective,
SeaAndPolarIce sends SeaLevel to LandFlood!
```

In addition to components and couplings, an SES can include specializations that provide alternative choices for components. For example, the SES for GreenHouseEffect of shown in outline in Figure 4, depicts a specializations rateOfIncrease for FossilFuelBurning with alternatives FastIncrease, SlowIncrease, and ZeroIncrease. Similarly, GreenHouseGasDilution is a specialization for GreeHouseGas with values FastDilution, SlowDilution, and ZeroDilution. Such specializations multiply to provide a combinatorial space of possibilities constituting the family of models generated by the SES. Appendix 2 contains description of some basic natural language SES statements.

Choice of alternatives from specializations is done in a process called pruning [6]. After pruning, a transformation algorithm creates the hierarchical coupled model corresponding to the pruned entity structure.

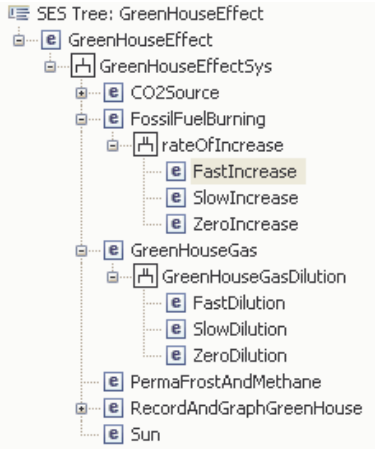


Figure 4 Outline of SES for GreenHouseEffect

4. DEVELOPING A SUITE OF MODELS

The process for developing, merging, pruning, and transforming an SES is illustrated in Figure 5. Note that before pruning, the SES components of a target SES are merged (recursively) to give the merged version of the target SES. The merged SES is then pruned and transformed to a hierarchical coupled model. For example, to create the merged SES for FloodIncreasing, the unmerged SES is merged with the component SESs (see Figure 2) GreenHouseEffect, SeanAndPolarIce, and LandFlood. where the latter is merged from components, LandFloodScape and LandFloodPlots.

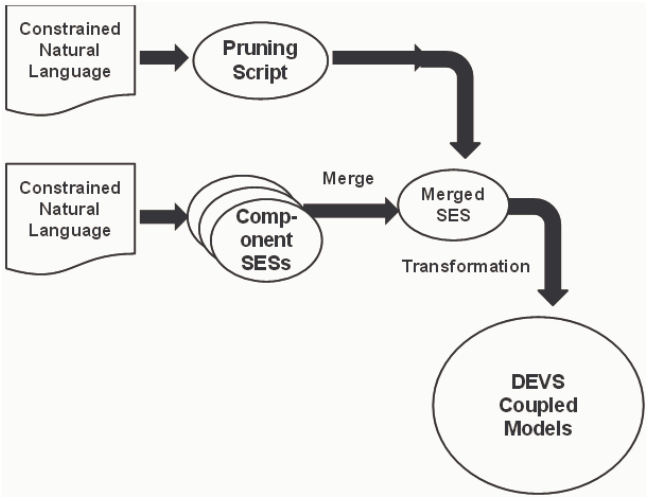


Figure 5 Developing, merging, pruning, transforming SESs

To develop a new family of models in a suite of models, you develop the target SES for it, looking for existing models that you can use as components. Such existing models include families of models generated by SESs as well as atomic models in the repository. An existing family of models generated by an SES becomes a component SES when you terminate the top down specification of the target SES at a leaf entity with the same name as this SES. This component SES will be merged into the target SES in the process illustrated in Figure 5. For needed components that are currently not available in the suite of models, you go through the same procedure, with the additional task of recursively developing the components in the manner of the target. Of course you may decide at any point to develop a model as an atomic model rather than as one generated by an SES.

5. MARKETPLACE OF MODELS

To summarize, in a suite of models, each component SES represents a family of models that can be pruned and

transformed to execute in a simulation. Component SESs can be merged to a new SES with the same compositional properties. This functionality leads to the concept of a marketplace of models. SES supports families of models for combinatorial generation of architectural alternatives for exploration and optimization. As an ontological framework, the SES supports composition of models drawn from one or more model repositories. Operations on SES objects such as merging ease development by maintaining coherence of shared constructs among modifiable components. Merging enables divide-and-conquer component-based development of suites of models. Another operation, called mapping, supports tailoring and restructuring of SES components for different objectives [5].

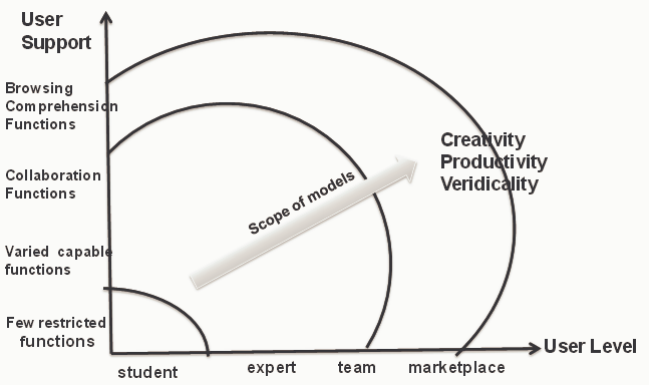


Figure x Supporting marketplace of models

The concept of marketplace of models extends the support for developing suites of models by using Web Services backed by Cloud Technology. MS4 Systems is developing environments to support the workflow development processes for a marketplace of models. The MS4 Store technology supplies the requisite Cloud-based model repositories. The MS4 Modeling Environment (Me) provides a range of support to enable evolution of users from neophyte to expert. For the student it provides an introduction to a simplified DEVS, Finite Deterministic DEVS. It then goes on to provide advanced support for developing full-fledged DEVS models in Java. For collaborative team developers, it provides the composition and integration facilities based on the SES described above. Going beyond team development, it provides support for market-oriented development of models. As illustrated in Figure 6 for the Global Warming suite of models, such support involves browsing the SES structures of models aimed at enabling comprehension of model content and functionality. Such comprehension is necessary to realistically enable a developer to acquire and re-use a simulation model developed by someone else.

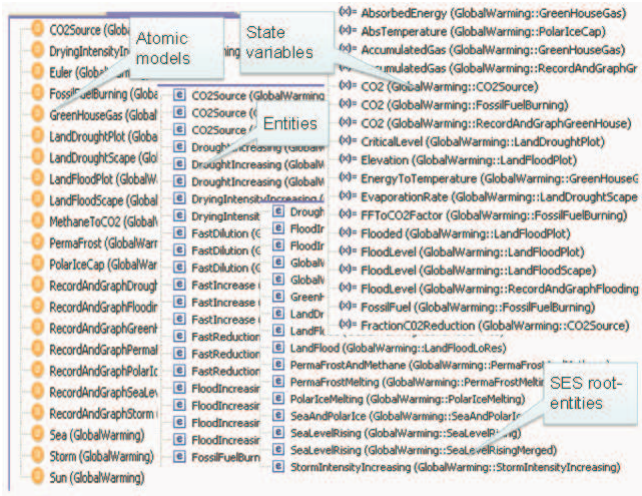


Figure 6 Browsing support provided by MS4 Me

6. CONCLUSIONS

This paper reviewed the System Entity Structure concept and its support for developing a suite of models in the MS4 Modeling Environment. We described the operation of merging SESs and its key role in the methodology for such development. As an ontological framework for modeling and simulation, the SES provides the basis for other advanced operations such as mapping and tailoring, in support of reuse, composition and integration. The framework also supports browsing and comprehension of model suite structure and behavior. Further research is needed to develop the most efficient and effective ways of generating views of SES entities, relationships, and variables. After construction, the suite of models can be hosted on Model Store, the cloud-based repository of models provided by MS4 systems as a basis for collaborative and market-oriented model development.

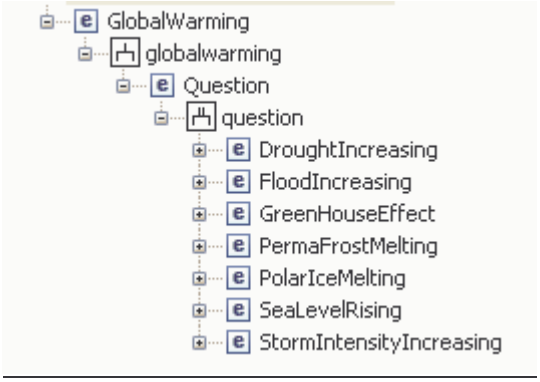
References

1. Kim, T., Lee, C., Christensen, E., Zeigler, B.: System entity structuring and model base management. Systems, Man and Cybernetics, IEEE Transactions on 20(5), 1013–1024 (1990)
2. Rozenblit, J., Zeigler, B.: Representing and constructing system specifications using the system entity structure concepts. In: Proceedings of the 25th conference on Winter simulation, pp. 604–611. ACM (1993)
3. Couretas, J. M., Zeigler, B. P. And Patel, U., 1999a, Automatic Generation Of System Entity Structure Alternatives: Application To Initial Manufacturing Facility Design. *Transactions Of The Society For Computer Simulation*, **16**.

4. Hagendorf O., Pawletta T. (2009) *A Framework for Simulation Based Structure and Parameter Optimization of Discrete Event Systems*.in Discrete-Event Modeling and Simulation: Theory and ApplicationsGabriel A. Wainer (Editor), Pieter J. Mosterman (Editor), CRC Press, 2010
5. Zeigler, B.P and Phillip Hammonds (2007), "Modeling&Simulation-Based Data Engineering: Introducing Pragmatics into Ontologies for Net-Centric Information Exchange", ", Academic Press, Boston,. 448 pages
6. Zeigler, Bernard P., Sarjoughian, Hessam S. Guide To Modeling And Simulation Of Systems Of Systems Series: Simulation Foundations, Methods And Applications Springer Pub. Co., pp. 330, 2013.
7. Verbraeck. Component-based distributed simulations. the way forward? In Proceedings of the 18th Workshop on Parallel and Distributed Simulation (PADS'04),pages 141–148, 2004.
8. M. R¨ohl and A. M. Uhrmacher. Composing simulations from xml-specified model components. In Proceedings of the Winter Simulation Conference 06, pages 1083–1090. ACM, 2006.

Appendix 1: Global Warming Suite of Models

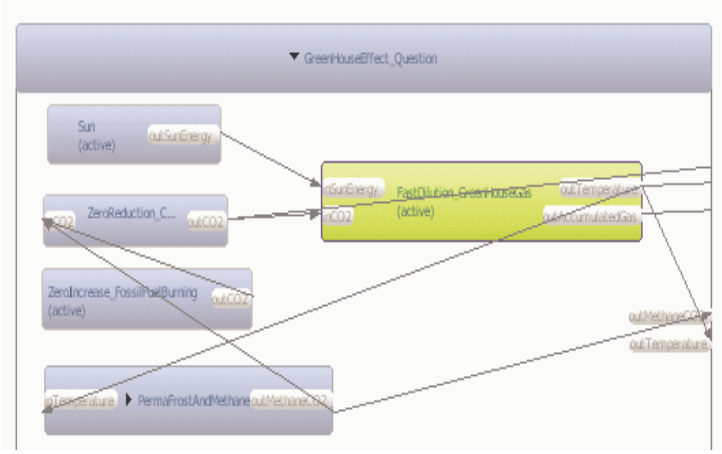
Each global warming model is addressed to a particular question as illustrated in the following breakdown:



Greenhouse effect Model

What is the greenhouse effect that causes warming of the earth?

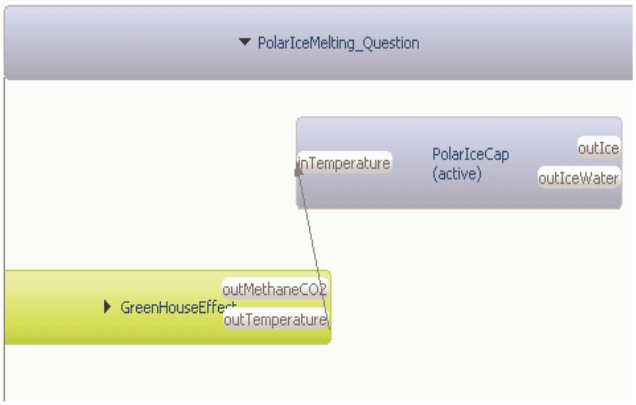
The greenhouse effect is the warming of the earth due to the accumulation of "greenhouse" gases in the atmosphere. Burning of fossil fuels such as oil and coal to power cars and generate electricity emits carbon dioxide into the atmosphere. The accumulated carbon dioxide traps the sun's energy reflected off the earth's surface, as if it were a greenhouse roof. The trapped energy warms the gases and raises the earth's temperature. A second source of carbon dioxide ($C0_2$) is gas released from permafrost in the polar region when it is heated due to the greenhouse effect – thus constitution a vicious feedback cycle.



PolarIceMelting Model

How does global warming cause increased melting of the Polar Ice Cap?

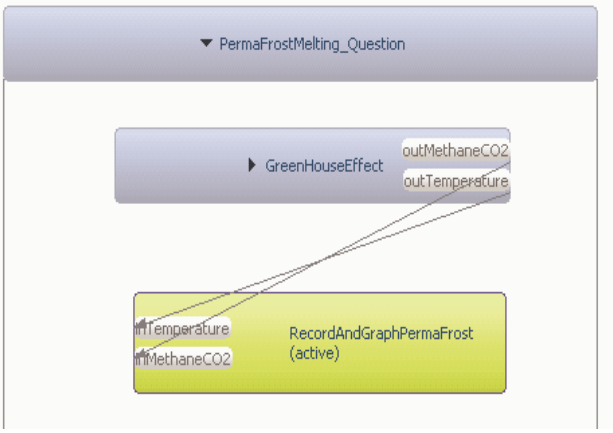
Increased melting of the ice at the Arctic Pole is caused by warming due to the Greenhouse effect.



PermaFrostMelting Model

How does the warming of the permafrost contribute to ever increasing global warming?

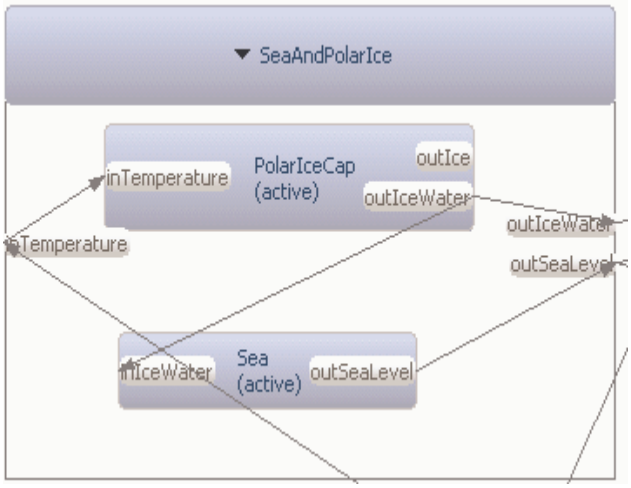
Melting of the permafrost (ground that is permanently frozen) is also caused by warming due to the Greenhouse effect. However, it also contributes to this effect since upon melting it also releases sequestered $C0_2$ to the atmosphere.



SeaLevelRising Model

How does the global warming contribute to the rising of the sea level?

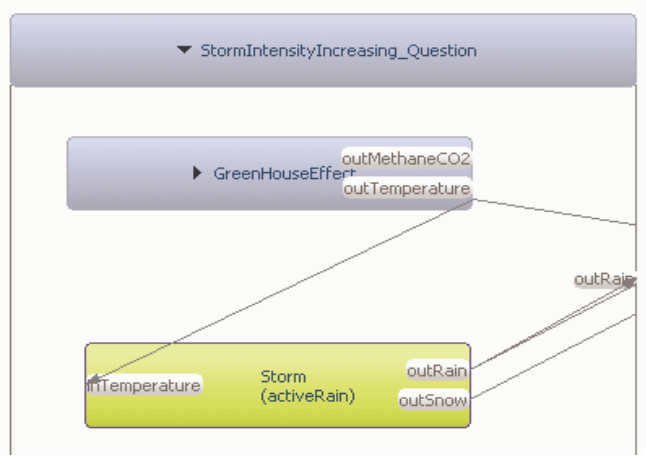
The level of the oceans' surface (sea level) is rising due to the Greenhouse effect as the ice of the Polar Ice Cap melts (caused by warming) and pours into the sea.



StormIntensityIncreasing Model

How does the global warming contribute to the increasing intensity of storms?

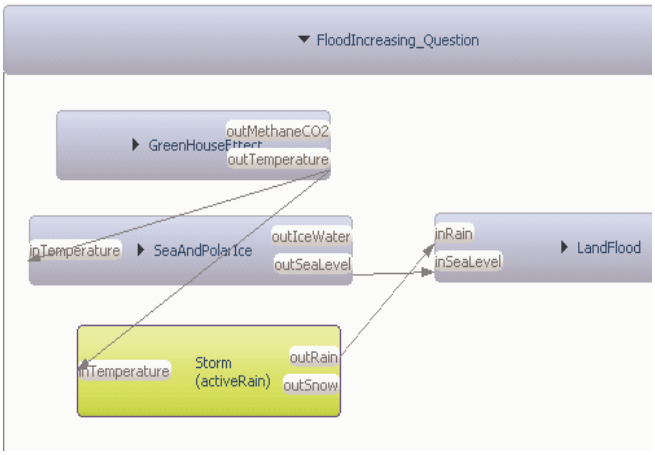
The intensity of storms is increased by the Greenhouse effect causing more severe and frequent precipitation (rainfall and snowfall.)



FloodIncreasing Model

How does the global warming contribute to the increasing incidence of floods?

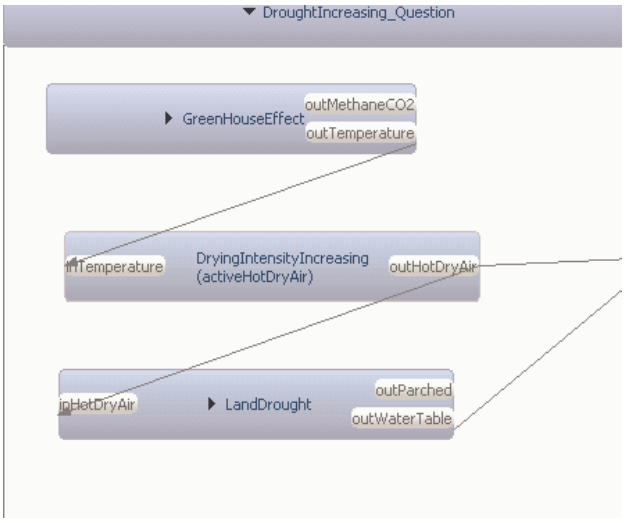
Floods are increasing in frequency and severity due to the sea level rising and storms growing in intensity both as a result of warming caused by the Greenhouse effect.



DroughtIncreasing Model

How does the global warming contribute to the increasing incidence of droughts?

Similar to the case of floods, droughts are also increasing in frequency and severity due to global warming. The greenhouse effect warms the atmosphere and dries the moisture in the atmosphere thus reducing the amount of water reach the earth's surface.



Appendix 2: Basic Statements for the System Entity Structure

The "Made of" statement

This statement tells how a modeled component is made of, or composed from, more basic components. Alternatively, it tells how the entity representing a component is decomposed into smaller entities using the aspect relation.
From the <perspective> perspective, the <coupled model> is made of <component1>, <component2>, …, and <componentN> !
Example: two aspects
From the floodHiRes perspective, LandFlood is made of LandFloodScape and LandFloodPlots!
From the floodLoRes perspective, LandFlood is made of LandFloodScape!

The "Sends" (Internal Coupling) statement

A series of these statements tells how the components of a coupled model are coupled. i.e., how their output and input ports are connected together.
From the <perspective> perspective, <component1> sends <Message> to <component2>!
or

From the <perspective> perspective, <component1> sends <outPort> to <component2> as <inPort>!

From the floodHiRes perspective, LandFlood sends Rain to LandFloodScape!
From the floodHiRes perspective, LandFlood sends SeaLevel to LandFloodScape!

The "Sends" (External Input Coupling) statement

A series of these statements tells how the input ports of a coupled model are coupled to input ports of its components
From the floodHiRes perspective, LandFlood sends Rain to LandFloodScape!
From the floodHiRes perspective, LandFlood sends SeaLevel to LandFloodScape!

The "Sends" (External Output Coupling) statement

A series of these statements tell how the output ports of a coupled model are coupled to output ports of its components.

From the <perspective> perspective, <component> sends <Message> to <coupled model>!

From the floodHiRes perspective, LandFloodPlots sends Flooded to LandFlood !
From the floodHiRes perspective, LandFloodScape sends FloodLevel to LandFlood!

The "Can be" (Specialization) statement

Specifies alternative choices for a component that can be made in pruning. A component can be though of as a place holder into which one of the alternatives can be "plugged."

<component> can be <alternative1>,<alternative2>, <alternative>, ... in <specialization>!

LandFloodPlot can be lowLying, seaLevelLying, or highLying in aboveGroundHeight!

The "Merge All" statement

Merges a set of SESs into a target SES, the set is the set of SESs found in the repository whose root entities exist as leaf entities in the target SES
MergeAll from <SES>
mergeAll from GreenHouseEffect.ses !