The Modelica Bond Graph Library

François E. Cellier
Institute of Computational Science
Swiss Federal Institute of Technology
ETH-Zentrum HRS H28
CH-8092 Zürich
Switzerland
FCellier@Inf.ETHZ.CH

Àngela Nebot
Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Jordi Girona Salgado, 1-3
Barcelona 08034
Spain
Angela@LSI.UPC.ES

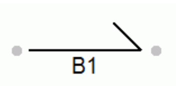
Abstract

Bond graphs offer a domain-neutral graphical technique for representing power flows in a physical system. They are particularly powerful for representing systems that operate in multiple energy domains, such as thermal models of electronic circuits, mechanical vibrations in acoustic systems, etc. A bond graph library was created for Modelica with graphical Dymola support. The library is presented in this paper. Applications from different domains are offered to document its use.

Keywords: bond graph; energy modeling; thermodynamic modeling; Biosphere 2

1 Introduction

A bond represents the flow of power, P, from one point of a physical system to another.

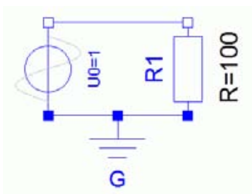


It is represented by a harpoon. There are two physical variables associated with each bond, an effort, e, and a flow, f. The product of these two variables represents the power:

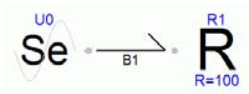
$$P = e \cdot f$$

In an electrical circuit, the effort variable is identified with the voltage, u, whereas the flow variable is identified with the current, i.

In the electrical circuit:



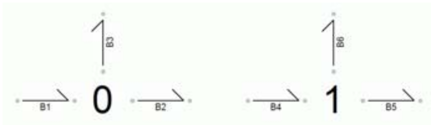
electrical power is being delivered from a voltage source to a resistor. The corresponding bond graph is:



where a source of effort, U_0, delivers the electrical power, $P_{el} = U_0 \cdot i$, to the resistor, R_1.

In our implementation, a third variable is also associated with each bond, a directional variable, d. This variable indicates the direction of positive power flow. It is encoded by setting $d = -1$ at the emanating bondgraphic connector and $d = +1$ at the receiving connector. The directional information is used in the computations associated with *junctions*.

Bond graphs offer two types of junctions, the 0-junction, and the 1-junction:



In a 0-junction, the efforts are set equal, whereas the flows add up to zero:

$$\mathbf{e}[2:n] = \mathbf{e}[1:n-1]$$
$$\mathbf{d}' \cdot \mathbf{f} = 0$$

In a 1-junction, the flows are set equal, whereas the efforts add up to zero.

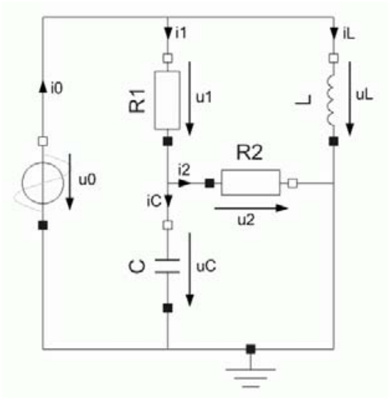
$$\mathbf{f}[2:n] = \mathbf{f}[1:n-1]$$
$$\mathbf{d}' \cdot \mathbf{e} = 0$$

Thus, the two junction types are duals of each other.

In an electrical circuit, the 0-junctions correspond to nodes, whereas the 1-junctions correspond to meshes. We are now able to translate the circuit diagram of an arbitrary electrical circuit into a corresponding bond graph.

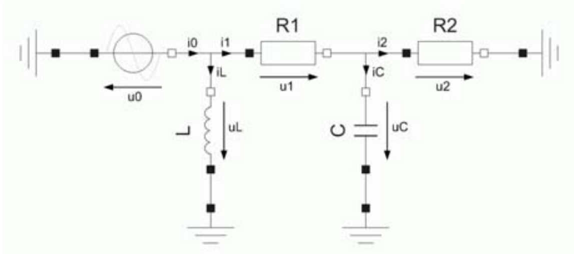
2 Bond Graphs of Electrical Circuits

Given the electrical circuit:

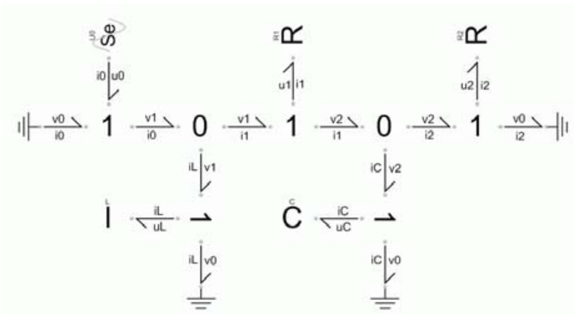


we can represent each node, except for the ground node, by a 0-junction, and we can represent each circuit element connecting two nodes by a 1-junction, from which the circuit element is suspended. The directions of positive power flows are chosen to coincide with the directions of positive current flow in the circuit diagram.

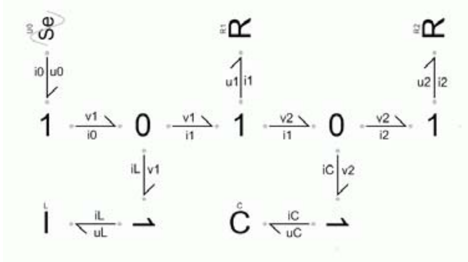
For simplicity, let us first redraw the circuit diagram with the ground node split into multiple separate nodes.



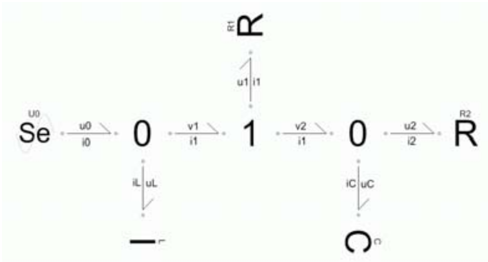
This circuit diagram can be translated directly into a corresponding bond graph:



Since the ground potential, v_0, is equal to zero, the bonds connecting to the ground don't carry any power. They can thus be eliminated.



Finally, junctions with only two bonds attached to them can be amalgamated away. Thus, the final bond graph can be drawn as follows:



3 Causal Bond Graphs

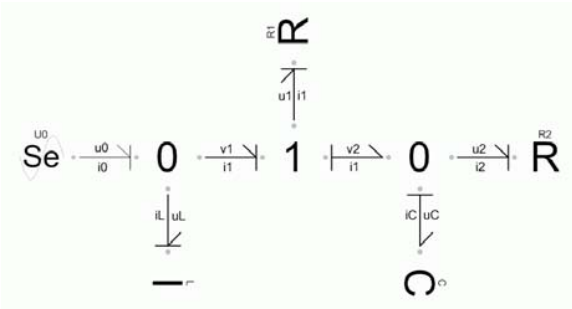
Since there are two physical variables associated with each bond, we need two equations to compute their values. It turns out that each end of the bond computes one of the two variables. We can mark the side that computes the flow variable by a short bar: the *causality stroke*.

The Modelica Bond Graph Library offers causal and a-causal bonds. Whereas the a-causal bonds were implemented as Modelica *models*, the causal bonds were implemented as *blocks*.

We recommend using causal bonds as much as possible. The causalities associated with bonds attached to sources are fixed. Since an effort source computes the effort, the causality stroke of its bond must be away from the source. Since 0-junctions are characterized by a single flow equation, there must be exactly one causality stroke at a 0-junction. Since 1-junctions are characterized by a single effort equation, there are exactly *n-1* causality strokes at a 1-junction.

Capacitors and inductances have preferred causalities. Since we like to end up with differential equations (integral causality), capacitors like to compute the effort, whereas inductors prefer to compute the flow. Thus the preferred position for causality strokes of bonds attached to capacitors is away from the capacitor, whereas the preferred position of causality strokes of bonds attached to inductors is at the inductor. The causalities of resistive elements are free.

In the case of the given circuit, the preferred causality of all bonds is fixed. The causal bond graph can be presented as follows:



We are now capable of reading out the causal equations from the bond graph. These are:

$$u_0 = f(t)$$
$$i_0 = i_L + i_1$$
$$u_L = u_0$$
$$di_L/dt = u_L / L$$
$$v_1 = u_0$$
$$u_1 = v_1 - v_2$$
$$i_1 = u_1 / R_1$$
$$v_2 = u_C$$
$$i_C = i_1 - i_2$$
$$du_C/dt = i_C / C$$
$$u_2 = u_C$$
$$i_2 = u_2 / R_2$$

There is no advantage of using an a-causal bond graph instead of a circuit diagram when modeling an electrical circuit. The two representations are totally equivalent to each other. However, there is a certain advantage of using a causal bond graph, since the equations describing the circuit can be read out of the causal bond graph directly in their causal form.

Of course, there is no need to ever use causal bonds in Modelica, as Modelica is perfectly capable of determining the computational causality of all equations on its own. Yet, we recommend using causal bonds as much as possible, as they help the modeler in analyzing his or her model.

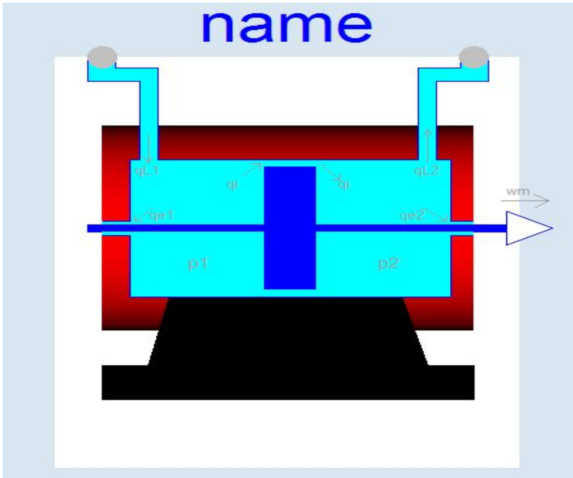
4 Algebraic Loops and Structural Singularities

When the mandated and preferred causalities of all elements do not lead to a single assignment of all causality strokes, the model contains one or several *algebraic loops*. A-causal bonds must be used wherever the causality assignment is free.

On the other hand, if not all preferred causalities can be satisfied, i.e., when the causality stroke of a bond attached to either a capacitor or an inductor is located at the incorrect end of the bond, the model contains a structural singularity, i.e., consists of a higher-index DAE system. Also in that case, a-causal bonds should be used to give Modelica a chance to reducing the perturbation index on its own.

5 A Hydraulic Motor Control System

We wish to model the following hydraulic motor by a bond graph:



In hydraulic bond graphs, it is customary to identify the pressure, p, with the effort variable, whereas the volumetric flow rate, q, is identified with the flow variable. The product of pressure and volumetric flow is the hydraulic power.

Due to the compressibility of the liquid, the change of pressure in each chamber is proportional to the difference between inflow and outflow. In terms of a bond graph, this looks like a capacitor attached to a 0-junction.

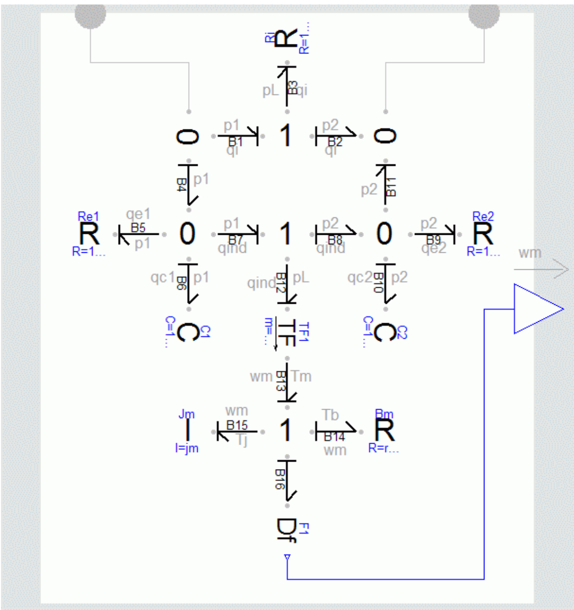
The flows q_i, q_{e1}, and q_{e2} are laminar leakage flows. They are proportional to the pressure difference. Thus, they can be represented as linear resistors.

On the mechanical side, power can be written as either force times velocity or torque times angular velocity. Among bond graph practitioners, it has become customary to identify the forces and torques with effort variables, and the velocities and angular velocities with flow variables.

Newton's law states that the change in velocity (or angular velocity) is proportional to the sum of all forces (or torques). In terms of a bond graph, this looks like an inductor attached to a 1-junction.

The two domains are coupled by a transformer, as the force on the piston (or the torque on the screw, depending on the geometry of the motor) is proportional to the difference between the pressures in the two chambers.

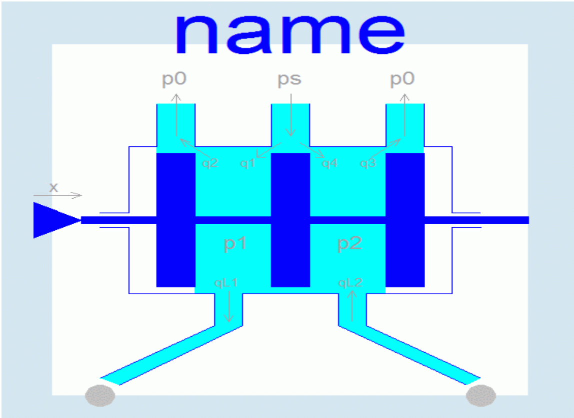
We are now ready to draw the bond graph of the hydraulic motor:



The two 0-junctions to the left and to the right represent the two hydraulic chambers with the pressures p_1 and p_2, respectively. Each of them has been pulled apart into two separate 0-junctions connected by a bond for graphical reasons. Same sex junctions neighboring each other can always be considered as a single junction. The two capacitors symbolize the compressibility of the liquid. The three resistors at the top half of the bond graph represent the leakage flows, one of which, q_i, is an internal leakage flow, whereas the others, q_{e1} and q_{e2}, are external leakage flows.

The transformer, *TF*, separates the hydraulic from the mechanical side. The inductor represents the inertia of the (rotational) screw, whereas the resistor represents the friction of the screw. The flow detector element, *Df*, detects the angular velocity, ω_m, of the screw. It converts the bond graph representation to a signal.

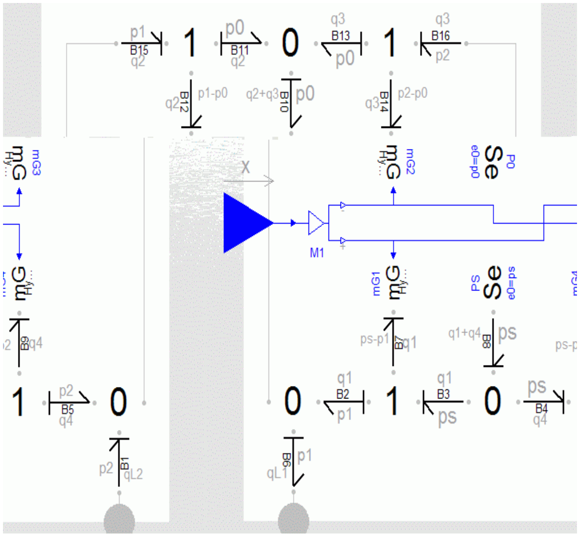
The hydraulic motor is controlled by a servo valve:



The inflow pressure, p_s, is the load pressure of the hydraulic motor. The outflow pressure, p_0, is the ambient pressure of the environment. The four valve flows, q_1, q_2, q_3, and q_4, are turbulent flows. Hence they are proportional to the square root of the pressure difference. In terms of a bond graph, they can be represented either as nonlinear resistors (*R*-elements) or as nonlinear conductors (*G*-elements). Since the causalities are those of a conductive element, we chose the latter representation to prevent Modelica from having to turn these nonlinear equations around symbolically.

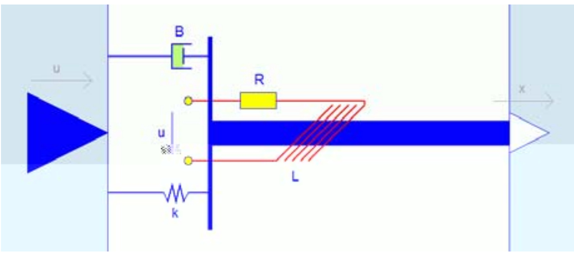
All four valve flows are modulated by the position of the tongue, *x*.

We are now ready to draw the bond graph of the servo valve:



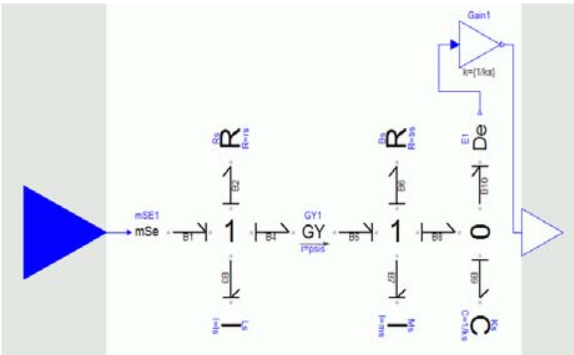
The tongue position, *x*, is an input signal. It influences the bond graph by means of modulation of the four hydraulic conductance elements.

We still need to model the motion of the tongue of the servo valve:



The tongue of the servo valve is an electromechanical converter. The input signal, u, modulates an effort source that generates a magnetic field in a coil. The magnetic field induces a mechanical force in the tongue that is proportional to the current through the coil. Thus, the converter can be modeled as a bond-graphic gyrator, *GY*. Whereas a *transformer* sets the output effort proportional to the input effort (and the input flow proportional to the output flow), the *gyrator* sets the output effort proportional to the input flow (and the input effort proportional to the output flow).

We are now ready to draw the bond graph of the device:

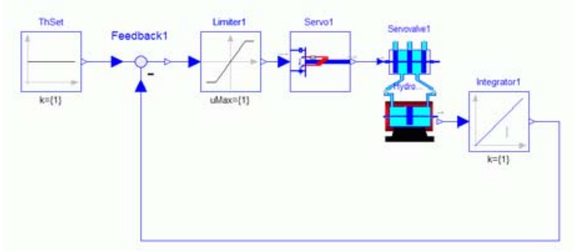


The modulated effort source translates the signal, u, to a power flow. The 1-junction to the left of the gyrator symbolizes the electrical mesh. The inductor here represents the coil, whereas the resistor represents the electrical resistance.

The gyrator converts the electrical power, $P_{el} = u_i \cdot i$, where u_i is the induced voltage, to mechanical power, $P_{mech} = f \cdot v$. The 1-junction to the right of the gyrator symbolizes the velocity of the tongue. The inductor here represents the mass of the tongue, the resistor represents the mechanical damper, and the capacitor represents the mechanical spring.

We could have attached a flow detector, *Df*, to the 1-junction to detect the velocity of the tongue. We could then have integrated the resulting signal to obtain the tongue position, x. Yet, we chose another route. The tongue position, x, is proportional to the spring force. Thus, we can use an effort detector, *De*, to detect the spring force. However, an effort detector needs to be attached to a 0-junction. To this end, an additional 0-junction was placed between the 1-junction and the capacitor.

We are now ready to model the control circuit:



From the outside, the control circuit looks like a regular block diagram. However, three of the blocks have been modeled by bond graphs internally.

6 The Thermal Budget of Biosphere 2 without Air-conditioning

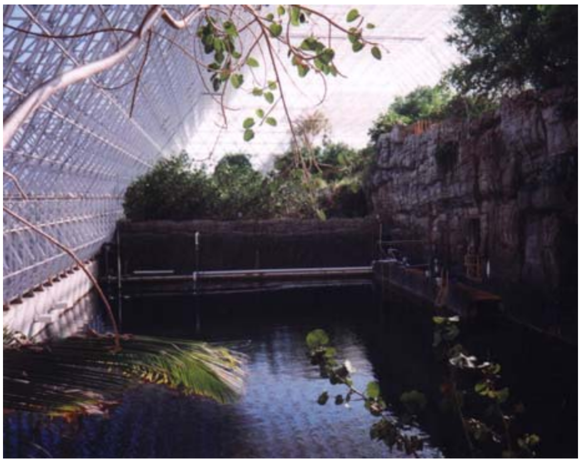
As a second example, we shall model the thermal behavior of Biosphere 2, an experimental research facility located in the vicinity of Tucson, Arizona, without air-conditioning by means of bond graphs.

Rather than using a bottom-up approach, as we did in the previous example, we shall this time around use a top-down approach.

Biosphere 2 was designed as a materially closed structure to investigate the ability of humans to survive in a materially closed structure for extended periods of time. The main idea was to investigate whether space colonies are feasible with today's technologies.

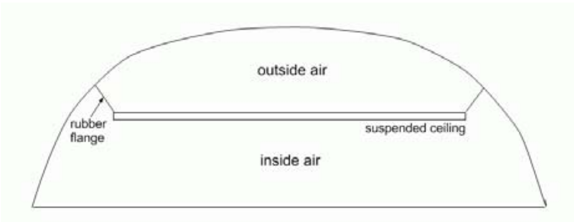
Biosphere 2 was constructed as a large glass building on 3 acres (12.000 m^2). The structure is held together by a metallic frame construction and is closed off by glass panels.

The structure contains a number of different biomes. The pyramidal structure to the right contains a tropical rain forest. The elongated structure to the left contains a pond, a savannah, saltwater marshes with mangroves, and a southwestern desert landscape.



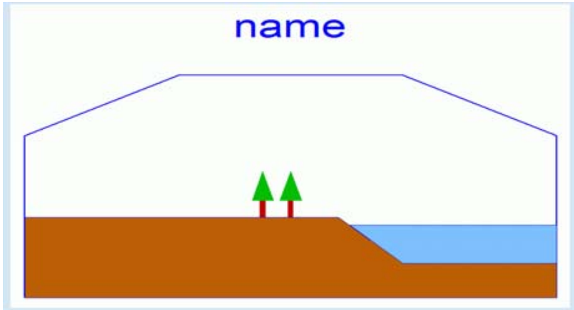
The air pressure inside Biosphere 2 is kept constant by two lungs, one of which is shown below:

The lungs operate as follows. A heavy cement ceiling is suspended from the dome by a rubber flange. The bottom part of the lung is inside the materially closed structure, whereas the top part is outside air.



When the temperature inside Biosphere 2 rises, the inside pressure increases as well. Consequently, the ceilings in the two lungs are lifted up, thereby increasing the total volume of Biosphere 2. In this way, the inside pressure remains the same as the ambient pressure irrespective of the temperature.

Although air-conditioning keeps the temperature and humidity values different in the different biomes, Biosphere 2 was modeled by us as a single structure with one inside temperature and humidity.



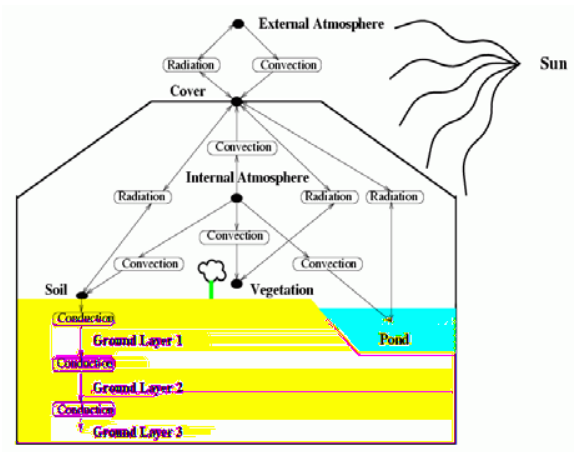
There are a number of different elements in that model: the inside air, the dome, the pond, the vegetation, and the soil, each of which are allowed to be at a different temperature. Only the inside air also contains humidity.

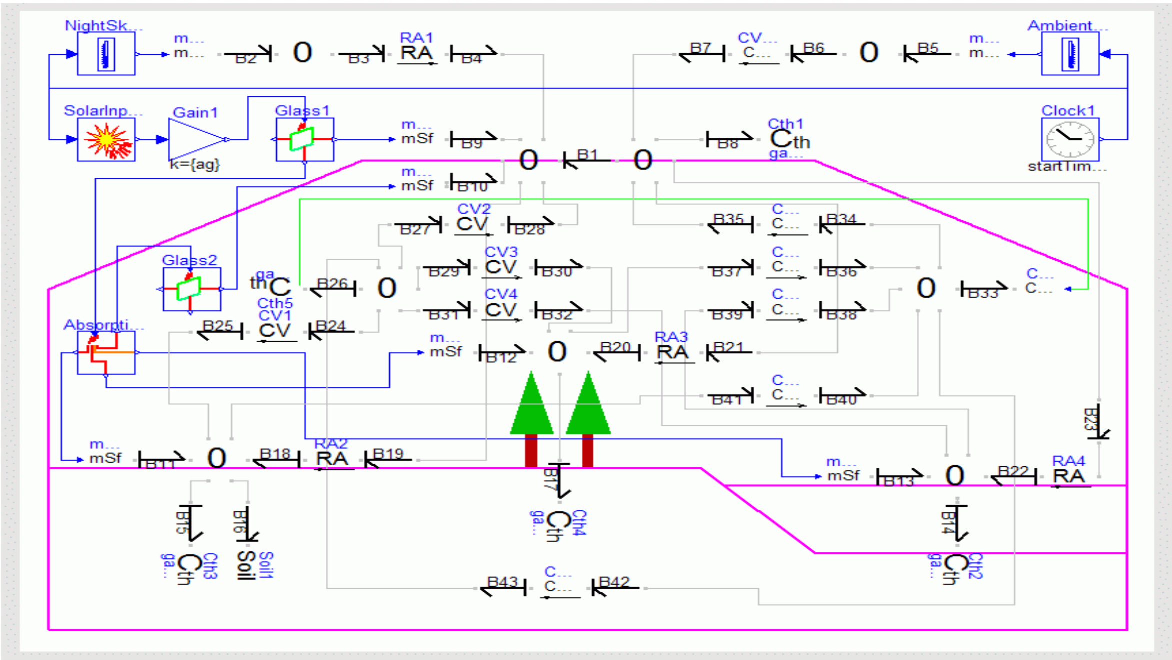
In a bond graph, thermal power, i.e., heat flow, can be written as the product of temperature and entropy flow. It is customary to identify the temperature with the effort variable, and the entropy flow with the flow variable.

Each of the five elements is represented as a 0-junction with a (non-linear) capacitor attached to represent the heat capacity of the element. Heat flows between the elements are represented as non-linear resistors modeling physical effects such as convection and radiation.

The inside air is represented by two separate 0-junctions, one modeling the temperature of the air, the other modeling its humidity. Non-linear resistors between the thermal and humidity junctions are used to model the effects of evaporation (conversion of sensible heat to latent heat) and condensation (conversion of latent heat to sensible heat).

A conceptual model of Biosphere 2 is shown below:





Each black dot represents a modeling element, i.e., a 0-junction with a heat capacity attached to it. The flows between these modeling elements are represented by two-port elements modeling the effects of conduction, convection, radiation, evaporation, and condensation.

The model starts in the upper right corner with the simulation clock. The ambient temperature and the apparent temperature of the night sky are computed by tabular look-up functions.

The temperature values are then converted to power flows by the use of modulated effort sources. Temperature sources are physically dubious, but it is okay to use them here, since the model doesn't contain any physical explanation as to how the environment reaches its temperature. The temperature values are simply being observed.

Since the dome is in physical contact with the outside air, convection takes place across the dome. Furthermore, the dome is exposed to diffuse radiation from and to the sky.

The 0-junction representing the dome was split into two separate 0-junctions connected by a bond for graphical reasons. The thermal capacitor attached to the 0-junction computes the temperature of the dome.

Biosphere 2 is also exposed to direct solar radiation. The *Solar Input* model, symbolized by the sun, computes the position of the sun in the sky, and thereby computes the total amount of direct solar radiation input reaching the Biosphere 2 structure.

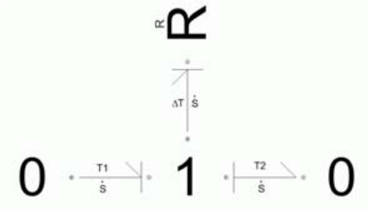
Since the different glass panels have many different orientations, it would have been computation intensive to calculate accurately the amount of radiation that gets transmitted, absorbed, and reflected by each of the glass panels. Thus, a much more global approach was taken. It is assumed that roughly 60% of the solar input gets transmitted across the glass panels, 20% gets absorbed by them, whereas the final 20% get reflected to the outside.

The $Glass_1$, *Absorption*, and $Glass_2$ models railroad the available solar input to the individual elements, where they arrive in the form of flow (entropy) sources. The vegetation, soil, and inside air absorb all of the arriving solar input. The pond absorbs some of it, and reflects the rest. The reflected solar input is partly absorbed by the inside air, and partly reaches the dome again from the inside, where it is partly absorbed, and partly transmitted back out.

Thus, a global balance approach was used to model the direct solar input. The end effect is that each of the 0-junctions representing the five different modeling elements has a modulated flow source attached to it that models the amount of direct solar input absorbed by that element.

7 Convection

Let us now look at the processes of convection between modeling elements. Since the air-conditioning was left out of the model, there are no forced flows. Thus, the convection is simply driven by temperature differences, i.e., by potential equilibration. This is a resistive phenomenon.

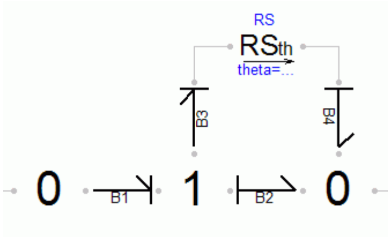


The two 0-junctions symbolize the two modeling elements that exchange heat among each other. They are at the temperature values, T_1 and T_2, respectively.

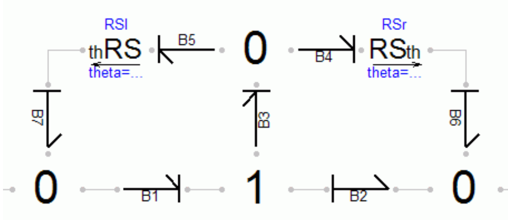
The 1-junction between them computes the temperature difference, ΔT, which drives the entropy flow.

The problem with this model should become evident at once. What happens with the power flow into the resistor? It may make sense to model with resistors in an electrical circuit, because we may not care about the entropy that is being generated by the resistor. However here, we are operating already in the thermal domain. Additional entropy is being generated by the resistor, and this entropy needs to be routed somewhere.

It has become customary to replace thermal resistors by resistive source elements, *RS*, and route the generated entropy to the nearest 0-junction. The so modified bond graph is shown below:

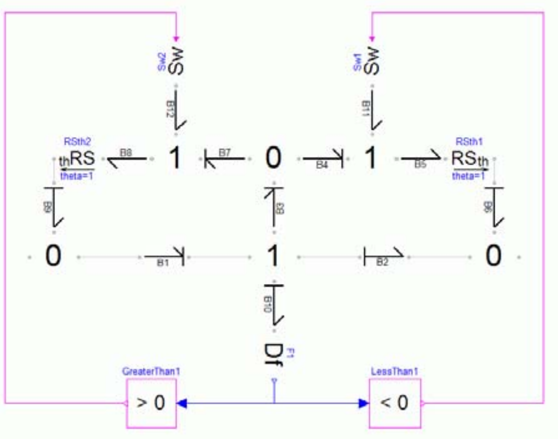


As convection is a symmetric phenomenon, we could alternatively route half of the generated entropy flow to the right and the other half to the left:



Finally, we may choose to route the generated entropy down-wind, i.e., if $T_1 > T_2$, all of the generated entropy flow is routed to the right, otherwise to the left.

To this end, we shall require a flow detector and two switch elements:



The bondgraphic switch element, *Sw*, has a Boolean input. If that input has a value of *true*, the switch is *open*, i.e., there is zero flow. In that case, the causality stroke is at the switch element. On the other hand, if the Boolean input has a value of *false*, the switch is *closed*, and in that case, there is zero effort. Thus by now, the causality stroke has moved away from the switch. Hence a-causal bonds must be used at the switches.

Since the 1-junctions must have *n-1* causality strokes, another bond must also change its causality. This has to be the bond that leads to the resistive source element, *RS*.

8 Conclusions

In this paper, a bond graph library has been introduced that was designed to be used with Dymola. Since bond graphs are a graphical modeling tool, it may be much less desirable to use this library with Modelica alone, i.e., in an environment that is based on an alphanumerical representation of models.

This is already the second presentation of the Modelica Bond Graph Library. An earlier paper [4] had been prepared for a conference on bond graph modeling. Thus, whereas the earlier paper had been prepared for an audience that knew a lot about bond graphs, but little if anything about Modelica and/or Dymola, the current paper was written for an audience that is expected to be knowledgeable about Modelica and Dymola, but probably knows little if anything about bond graphs.

An earlier presentation of the Biosphere 2 model was published in [5]. The model presented in that paper had been developed using a much earlier version of Dymola, prior to the design of Modelica. At that time, a strictly alphanumerical version of a bond graph library had been used [1].

Bond graphs offer a fairly low-level interface to modeling physical systems. Thus, bond graphs should be used hierarchically in the context of complex systems [2]. The Biosphere 2 model demonstrates how bond graphs can be hierarchically structured. The hydraulic motor example demonstrates how bond graphs can be hidden inside other modeling metaphors, such as block diagrams.

The primary strength of bond graphs is their domain independence. For this reason, bond graphs are particularly suitable for the description of physical systems that operate in multiple energy domains. Energy conversions can be modeled easily and conveniently using transformers and gyrators.

As with any other modeling paradigm, there is nothing unique about bond graphs. Every single one of our models could have been developed using other modeling paradigms as well. Modeling paradigms offer a means for modelers to organize their knowledge about the physical systems they wish to describe. Some researchers will find bond graphs a convenient way to organize their knowledge, whereas other researchers won't. To us, bond graphs have become the ultimate tool for understanding the basic principles covering all of physics [3].

References

[1] Cellier, F.E. (1991), *Continuous System Modeling*, Springer-Verlag, New York, ISBN: 0-387-97502-0, 755p.

[2] Cellier, F.E. (1992), "Hierarchical Non-Linear Bond Graphs: A Unified Methodology for Modeling Complex Physical Systems," *Simulation*, **58**(4), pp. 230-248.

[3] Cellier, F.E. (1995), "Bond Graphs: The Right Choice for Educating Students in Modeling Continuous-Time Physical Systems," *Simulation*, **64**(3), pp. 154-159.

[4] Cellier, F.E. and R.T. McBride (2003), "Object-oriented Modeling of Complex Physical Systems Using the Dymola Bond-graph Library," *Proc. ICBGM'03, 6th SCS Intl. Conf. on Bond Graph Modeling and Simulation*, Orlando, Florida, pp. 157-162.

[5] Nebot, A., F.E. Cellier, and F. Mugica (1999), "Simulation of Heat and Humidity Budget of Biosphere 2 without its Air Conditioning," *Ecological Engineering*, **13**, pp. 333-356.

Biographies

François E. Cellier received his B.S. degree in electrical engineering from the Swiss Federal Institute of Technology (ETH) Zürich in 1972, his M.S. degree in automatic control in 1973, and his Ph.D. degree in technical sciences in 1979, all from the same university. Dr. Cellier joined the University of Arizona in 1984 as associate professor, where he is currently a full tenured professor of Electrical and Computer Engineering. Dr. Cellier's main scientific interests concern modeling and simulation methodologies, and the design of advanced software systems for simulation, computer aided modeling, and computer-aided design. Dr. Cellier has authored or co-authored more than 200 technical publications, and he has edited several books. He published a textbook on Continuous System Modeling in 1991 with Springer-Verlag, New York. He served as general chair or program chair of many international conferences, and serves currently as president of the Society for Modeling and Simulation International.

Àngela Nebot received licentiate and Ph.D. degrees in computer science from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 1988 and 1994, respectively. From 1988 until 1992 she was with the Institut de Cibernètica, holding a pre-doctoral research grant from the government of Catalunya, and completing the doctoral courses in software and artificial intelligence. She joined the department of Llenguatges i Sistemes Informàtics in 1994 as an assistant professor, and since March 1998, she has been associate professor in the same department. She currently belongs to the soft-computing group of the UPC. She is an associate editor of the journal Simulation: Transactions of the Society for Modeling and Simulation International, and serves as a reviewer for other international journals such are the International Journal of General Systems, Neurocomputing, Artificial Intelligence in Environmental Engineering, and Artificial Intelligence Communication. Her current research interests include fuzzy systems, neuro-fuzzy systems, genetic algorithms, simulation, and e_learning.