

SUSTAINABLE MANUFACTURING ANALYSIS USING AN ACTIVITY BASED OBJECT ORIENTED METHOD

Yuriy Romaniw
Bert Bras
Tina Guldberg

Georgia Institute of Technology

Copyright © 2009 SAE International

ABSTRACT

This article begins by describing the need for a new method and tool for performing a sustainability assessment for manufacturing processes and systems. A brief literature survey is done to highlight the major existing methods and tools, their function, and their shortcomings. The article goes on to describe the general approach of the method before describing a computer aided tool that has been developed to implement the method. The article concludes with a walk through of a generic use case that describes where such a method would be useful and how such a tool would be implemented.

INTRODUCTION

In a modern world that is more aware of the scarcity of natural resources, it is becoming increasingly important to not only create and operate sustainable products but also be able to quantify and compare alternatives to best choose an optimal solution. A life cycle assessment (LCA) is one means of performing such a sustainability analysis. An LCA quantifies certain indicators throughout the entire life of a system. This type of assessment is known as a "cradle to grave" assessment. With respect to sustainability analysis, an LCA will focus on environmental impact indicators or resource consumption indicators. Such indicators can include, but are not limited to fuel consumption, raw material consumption, energy usage, emissions, and waste production. Current LCA tools generalize characteristics of a system during various stages of life that are based on a combination of empirical and

theoretical information. Existing LCA's, however, may be too general for parties that are interested in a specific stage of life. Of particular interest is the manufacturing stage of life. Other stages of life for a system, such as use phase and resource harvesting phase, have been well documented, but there is room for improvement when it comes to a sustainability analysis of the manufacturing stage of life. This article will focus on outlining a methodology and a tool that can help perform more detailed sustainability analysis of a manufactured part during the manufacturing phase of life.

MOTIVATION AND OBJECTIVE

Current LCA tools and methodologies are lacking in manufacturing phase information. To many industries that produce and sell products the manufacturing phase of life is the keystone to their operations. Industries, such as The Boeing Company may be able to perform generalized LCA's on how efficient their planes are during use phase, but Boeing is the primary operator of their products in the use phase. Boeing is not primarily involved in the design and manufacture phases of life for their products. Since the manufacturing phase of a products life defines such a large portion of this industries' operations, it is critical to have a way to perform a sustainability analysis on what it costs to manufacture a product, be it a plane, a car, or any other good.

The objective of this paper is to outline a methodology that can meet certain standards in helping carry out a more complete sustainability analysis of a manufactured part during the manufacturing phase of life. The method describes a model that is used to

perform such an analysis. This model must meet certain criteria:

1. The model must be flexible so that it can represent multiple different manufacturing processes
2. The model must be able to not only represent a system, but also execute simulations of that system
3. The model must be easy to use and modify, so that a user can readily define a system and easily interpret results

To meet the above requirements, a computer aided, object oriented, activity based costing model will be developed. An object oriented approach uses patterns of common elements to define classes that each have similar attributes and behaviors. Once properties have been defined, they do not need to be redefined every time a user creates another model. The user need only create instances of certain classes of elements, having those instances inherit properties and behaviors according to their base classifier. A computer aided approach will help with visualization and bookkeeping should the models get complex. An activity based costing (ABC) approach provides an underlying structure as to how the analysis would be carried out.

APPROACH

The model will be created in software called SysML (Systems Modeling Language). SysML is a plug-in for MagicDraw UML (Universal Modeling Language) and uses a graphical object oriented method to generating system models. To execute a simulation of a model, a SysML plug-in called ParaMagic, developed by InterCAX, which extracts data from the SysML model and performs mathematical computations in Wolfram Mathematica before updating the model with results. SysML is by no means the only software capable of creating such a model, it is simply the software tool used in this particular representation of the activity based, object oriented model (ABOOM) approach to sustainability assessments. Any software with similar capabilities should work. Additional details about the capabilities of SysML will be discussed later.

LITERATURE REVIEW

Nien-Lin Hsueh, et. al. [12] indicated that an object oriented approach is appropriate for constructing a model where repeatable patterns can be identified. Such repeatable patterns exist in manufacturing systems and are utilized by software, such as Computer Aided Manufacturing (CAM), Computer Resource Planning (CRP-II), and even more detailed Enterprise Resource Planning (ERP) software that broadens the scope of what is considered in the manufacturing process. A concept called Group Technology seeks out such similarities in order to help organize manufacturing system architecture. Group Technology lends itself greatly to object oriented modeling, because "the similarity in the characteristics of similar parts suggests that benefits can be obtained by classifying and coding

these parts into families." [13] Zhou Xiaoming and Du Pingan [21] outline an object-oriented approach to assembly sequence planning, showing that processes and not just parts can be commonly defined and classified. Hossam A. Gabbar [9] used an object-oriented approach to generate a design model for sustainable energy production and tracking of energy supply through a system. Gabbar created various classes that characterized energy, energy production methods, and other elements by generalizing common attributes amongst those elements. Freidenthal [8] indicates the benefits of using an object-oriented approach to systems engineering by showing that an element does not change, even when viewed from different viewpoints. He argues that an element can be created as an object, and various designers can modify that object's attributes in order to describe that object from various perspectives in the design stage.

There have been multiple models that strive to assess an environmental impact or compute an ecological impact indicator of an element of interest. Software, such as EcoIndicator '99 and SimaPro, assign attributes to parts and use these attributes to compute a value that indicates the environmental impact of that part. Several companies, such as General Electric, have their own online tools that can give a quick and dirty assessment of impacts based on selection of appliances. Environmental management through activity based costing utilizes patterns to model interaction between three element classes; Cost Objects, Activities, and Resources Pineda-Henson and Culaba [15] analyze the sustainability of a semiconductor manufacturing process by integrating a traditional life cycle assessment with a decision making process. Combining traditional life cycle assessment metrics with the decision making criteria helped produce a tool that determines how sustainable a particular manufacturing process is. Emblemssvag and Bras [7] use an activity based approach to a product life cycle assessment.

Sustainability assessments tend to use ISO 14000 codes and regulations. Bennett and James outline a detailed list of many physical values and how they are used to compute and indicator. Pineda-Henson and Culaba as well as Emblemssvag and Bras [7,15] refer to ISO 14000 regulations, but also refine the list of physical quantities to reflect their particular approach. This model outlined in this article will not have a comprehensive list of physical quantities that lead to an indicator. As an initial proof of concept, this article will only look at emissions of carbon dioxide, NO_x, and SO_x, as well as quantify how much fuel and resources were consumed during the manufacture of a hypothetical part. The model will ultimately be expanded to include more physical quantities, as well as a means to compute an indicator. Since the model presented uses an object oriented approach, adding detail, such as additional physical quantities, is straightforward. The new value types would be created in the appropriate element, and the appropriate behaviors need only be defined once.

SYSML TERMINOLOGY AND NOMENCLATURE

Though SysML is a very versatile and comprehensive software, this model will only use a limited number of features. This may change as the model is expanded and reviewed.

The fundamental element that will be used is called a “block.” A block can be used to represent any physical element or abstract idea, depending on the model. In this case, a block will be used to define objects and manufacturing operations. A block can contain many attributes, but the three primary attributes will be “references,” “values,” and “constraints.” References are elements whose information is referenced by a parent block, but are not contained within a block. References are themselves blocks. Values are numerical quantities that are characterized by “value types.” Value types define the dimension and unit of a particular quantity. Constraints are similar to blocks, but represent mathematical relationships. Constraints have parameters that are related through a constraint equation. These parameters are value properties of a block. Blocks can also have many kinds of behaviors. This model will focus only on parametric behaviors that show how values amongst different blocks relate through mathematical constraints. Additional behaviors may be defined later as the model is expanded and reviewed.

“Instances” are, as the name suggests, specific examples of a block. A block is used to classify elements, while instances create unique examples of that class. Instances are what will store specific numerical quantities for values and they will represent unique elements in the model. A user would primarily be manipulating instances in the model, having had the blocks’ attributes and behaviors predefined.

When elements appear graphically, their metaclass will appear in guillemets. For instance, when a block is represented, <<block>> will appear at the top of the elements. Similarly, value type will appear as <<value type>>, constraints as <<constraints>>, etc.

Camel case will be used as the primary writing style. This will assist plug-ins parse through the model, and will help maintain clarity in naming. The name of a class will begin with a capital or uppercase letter. For instance, let us say there is an element class called “manufacturing machine.” This would appear as ManufacturingMachine in camel case. Names of attributes in a block will begin with a lowercase letter. Let us say a block owns an element called a drill press that is classified as a manufacturing machine. The words drill press would appear as drillPress. The combined attribute will appear as drillPress:ManufacturingMachine, which can be read as “an element called a drill press that is of class manufacturing machine.” Instances will appear with an underline. For example, a Ryobi 12 inch drill press that is an instance of a Manufacturing Machine would appear as Ryobi12inDrillPress:ManufacturingMachine, which can be read as “an instance of the class manufacturing machine that is called Ryobi 12 in drill press.” The name drillPress of the element appears only in the attributes of the block that contains the drill press, not in the instance

name. The instance, however, can be used to fill the drillPress attribute of the block.

When an element name is given beginning with a capital letter or in camel case, then this is meant to refer to a particular element of that designation within the model. If the element name is not capitalized or in camel case, it will be assumed it is a general usage of the word outside of the model. For instance: “ManufacturedPart” refers to an element within the model, while “manufactured part” is the general reference to a part that is manufactured, outside of the scope of the model.

ACTIVITY BASED COSTING METHOD

Traditional ABC defines an element that is called an “object.” Objects are defined by or consume “activities,” which in turn consume “resources.” This can be seen graphically in Figure 1.

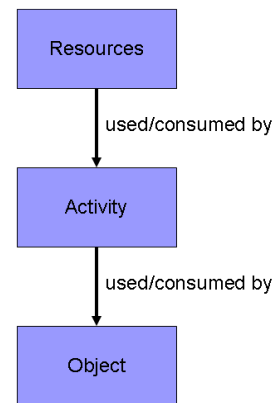


Figure 1: Traditional activity based costing breakdown

In this model, an object will be a manufactured part. Defining this part is a series of manufacturing operations that will function as activities. These operations will consume resources, which will be fuels, energy, materials, etc.

The traditional ABC structure is modified slightly in this model. The first key change is that activities can not only consume or use resources, but they can also produce waste. The second major change is that an object may contain other objects as well as activities, such as an assembly containing multiple parts and an assembly activity that joins them together. This allows for multi-scale system representation. The final significant change is that resources can, in some cases, be defined as objects that are produced by some sort of process described in terms of activities. An example of this would be a material resource like aluminum. Aluminum is refined (activity), and that refining process consumes resources.

Since this model represents a manufactured product, a naming convention will be used. The following elements will always be referred to with a capital or uppercase letter. An Operation is akin to an activity, and it can consume Resources and it can produce Waste. A ManufacturedPart is a fundamental manufactured element that can be defined completely in terms of Operations. A SubAssembly can contain multiple

ManufacturedParts and Operations that define how those parts are joined. Similarly, an Assembly contains SubAssemblies and Operations, while a Product contains SubAssemblies and Operations. Ultimately, a Product can be decomposed into a long list of Operations, each consuming Resources and producing Waste. This is shown graphically in Figure 2. Note that the words “Resources” and “Waste” are omitted, and are replaced with an “in” arrow, representing Resources, and an “out” arrow, representing Waste. The word Operation is abbreviated as “Op” and ManufacturedPart is abbreviated as “Part.”

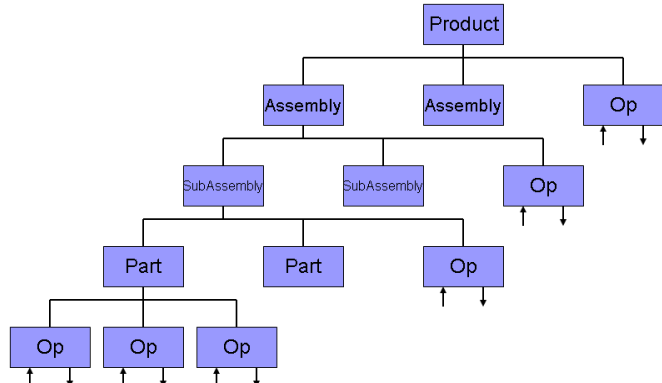


Figure 2: Structural breakdown of model elements

In addition to the above changes, there will be several other elements that are added to the model. The most common one will be an “actor.” Actors are contained in Operations and define at what rates Resources are being consumed and at what rate Waste and higher level objects such as Parts are being produced. Other minor changes will be either self-explanatory, or will be described when they appear in the model.

THE MODEL

CURRENT MODEL SCOPE

The model for this paper will focus on defining Parts and the Operations that describe them. The manufacturing phase of life for a ManufacturedPart will be the primary concern. Higher levels of the model will not be addressed in this paper and will be discussed further in the Future Work section. Several environmental impact quantities will be used, but this list is limited for now. This list includes consumable resources, such as fuels, raw materials, and energy, as well as waste such as carbon dioxide emissions, NOx emissions, and sulfur dioxide emissions.

MODEL STRUCTURE

There are three primary parts to this model. They are the Base Classifier Library (BCL), the Particular Element Library (PEL), and the Particular Instance Model (PIM). The BCL will be defined ahead of time within the tool and will be constant for all models. The PEL will be defined once by a particular user of the tool and will be referenced by the actual model. The PIM

will contain the final functional model and will be what the user will primarily be concerned with.

THE BASE CLASSIFIER LIBRARY

The BCL contains all of the generalized classes of elements that will be contained in the model. It will define all the attributes and behaviors of those classes, as well as any other classifier information that may be necessary. First, we define the Operation. All Operations will consume a quantity of fuel and a quantity of energy. They will all produce carbon dioxide, NOx, and sulfur dioxide emissions. This is shown graphically below.

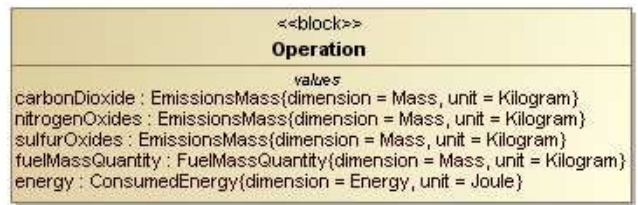


Figure 3: Definition of an Operation Class

These values are defined by value types that are shown in Figure 4.

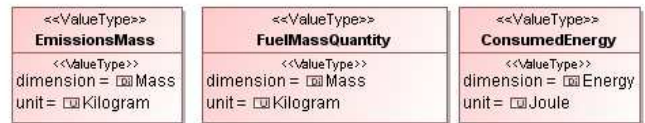


Figure 4: Value types used in Operation

This definition of Operation is general enough to capture some critical information, but it is too general for the purposes of this model. Therefore, three types of Operations will be defined, each having unique properties, but all being generalized by the definition of Operation in Figure 3.

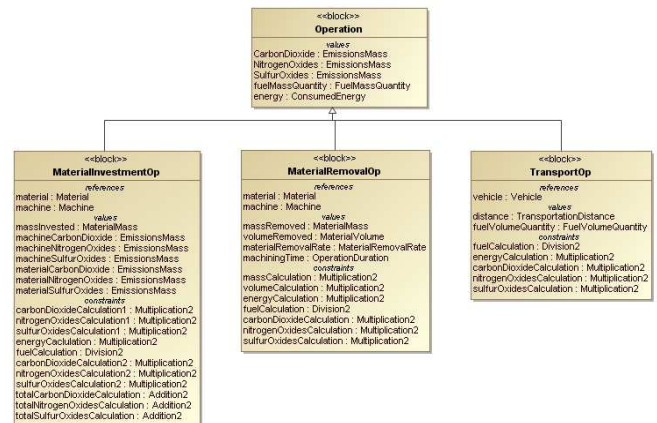


Figure 5: Specific Operation types

The three new Operation types that have been defined are material investment, material removal, and transportation operations. An example of a material investment operation is casting a ManufacturedPart.

This operation consumes a material resource and has a value of material mass invested *in addition to* of the values defined in Operation. Similarly, a material removal operation defines an actor (machine) and the material resource being consumed, as well as some physical properties of the operation, like mass removed, volume removed, material removal rate, and machining time. A transportation operation has a vehicle as an actor, and adds values of transportation distance and quantity of fuel consumed as values.

The actors of class Machine and Vehicle are defined in Figure 6 along with the definition of Fuel. Notice that each of the actors consumes a resource (Fuel) and then has a value associated with it defining at what rate the fuel is consumed. For Vehicles, this may be a miles per gallon value, or liters per 100 kilometers value, while it would be watts for a machine. Note that electrical energy is considered a Fuel. The class Fuel defines how much of a particular emission is produced per quantity of fuel, as well as how much energy is released in a unit quantity of fuel. New value types defined here are defined in the model, similarly to Figure 4, but are not explicitly shown in this paper.

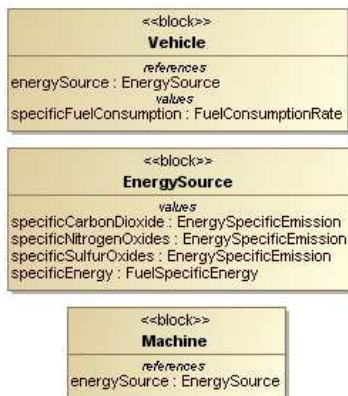


Figure 6: Definition of actors and the resource they consume

Also shown in Figure 5 is the resource of class Material, defined below. A Material is said to have associated with it a RefiningMethod that indicates what Resources and Waste were produced in order to produce a unit quantity of that Material. In this way, some information from the beginning of life of the product is captured. This is done for Material resources because some materials, such as aluminum, have an extremely high environmental cost to refine. These impacts are shown so that they may be compared to manufacturing phase costs.

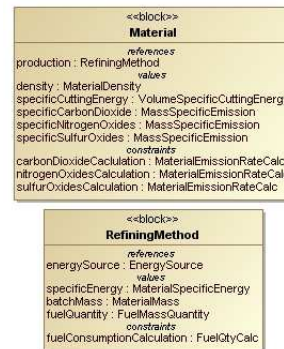


Figure 7: Material resource definition along with Refining Method

Now that the attributes of an Operation have been defined, a Part is defined as containing Operations. The multiplicity here is defined by the brackets. For instance, a Part contains one or more ([1..*]) ManufacturingOp's. Many new values are also added to track and tabulate various Resource and Waste information about the Part. These will not be described in detail here, but are assumed to be self-explanatory from their definition.

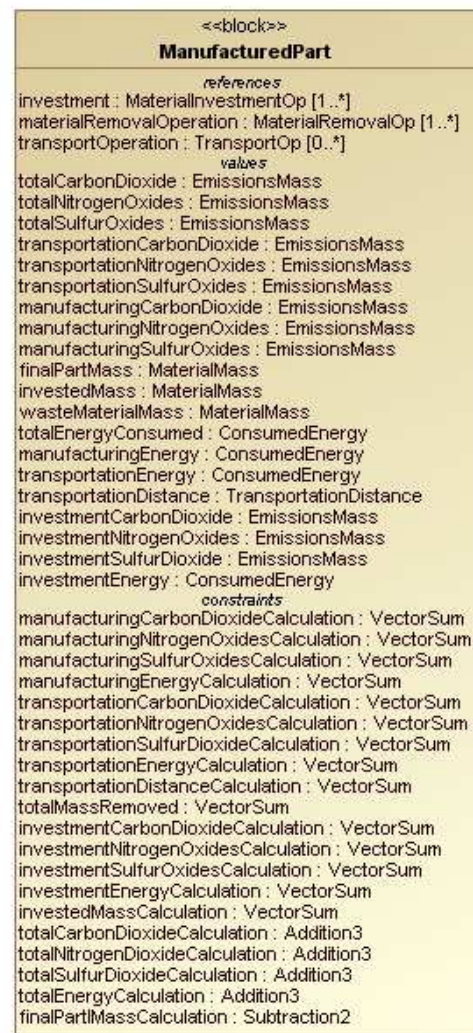


Figure 8: Definition of a Part

In the above figures, there are also a series of constraints listed that have not been mentioned yet. These constraints are mathematical relationships that define how the values in each of the elements are computed based on other values. There is a parametric diagram that shows how these relate for each of the elements above. An example of a parametric diagram for a ManufacturedPart is given below. Parametric diagrams tend to be large, so they will not be discussed in detail in this paper.

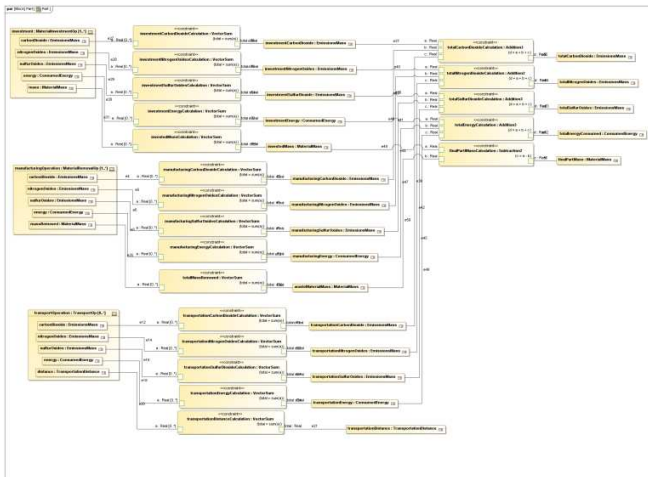


Figure 9: Parametric diagram for a Part

Since parametric relationships have to be explicitly defined, this adds a level of adaptability to the model that other LCA tools do not provide. The model is completely transparent, in that all the calculations are shown in one way or another. There are no “black box” calculations in this model. This also allows a user to trace the origin on a value, which can be useful when the source of some information is desired. Furthermore, explicit parametric diagrams allow a user to adjust the way in which values are calculated in order to create a model that more closely matches the system being modeled. For the most part, parametric diagrams would be defined once and not modified after that.

The above defines the basic structure of a BCL. All of the following elements that are defined are instances of the classes defined in the BCL. The complete BCL for this model is not shown above, but is contained within the model used in this paper. As indicated before, a user would not modify the BCL in most cases. The BCL defines the “behind the scenes” mechanics of the model, but is made available in the event that a user would want to modify the structure to more closely match a system, or to trace the origin on an attribute or behavior.

THE PARTICULAR ELEMENT LIBRARY

Certain elements can be defined ahead of time and do not change from model to model. The elements can be universal, but are generally particular to a specific user. These elements include, but are not limited to, Fuel properties, Material Properties, or certain actors that a particular user has ready access to. For instance, the properties of auto gasoline do not change from

model to model, and the same is true for materials like aluminum. A particular user may also have a list of instances of available Vehicles with their fuel consumption rates predefined.

The PEL is essentially an LCA inventory library. Much of the information in the PEL can come from existing LCA inventory databases. The rest of the information comes from the particular user’s inventory of available actors or resources. The reason the PEL is called a “library” and not an “inventory” or “database” is because each of the instances is an object that can be moved and manipulated independently. The data stored in these instances is always associated with that instance. This is unique from the traditional database or inventory where information is stored in spreadsheets and lists. An abbreviated example of such a library for Fuels is given in Figure 10.

<pre> <<block>> AviationGas : Fuel specificCarbonDioxide = "0.000000000359391"(dimension = EnergySpecificMass, unit = KilogramPerJoule) specificEnergy = "43700000"(dimension = MassSpecificEnergy, unit = JoulePerKilogram) </pre>
<pre> <<block>> DistillateFuel : Fuel specificCarbonDioxide = "0.000000000438279"(dimension = EnergySpecificMass, unit = KilogramPerJoule) specificEnergy = "48500000"(dimension = MassSpecificEnergy, unit = JoulePerKilogram) </pre>
<pre> <<block>> JetFuel : Fuel specificCarbonDioxide = "0.00000000041304"(dimension = EnergySpecificMass, unit = KilogramPerJoule) specificEnergy = "43200000"(dimension = MassSpecificEnergy, unit = JoulePerKilogram) </pre>
<pre> <<block>> Kerosene : Fuel specificCarbonDioxide = "0.000000000421695"(dimension = EnergySpecificMass, unit = KilogramPerJoule) specificEnergy = "43100000"(dimension = MassSpecificEnergy, unit = JoulePerKilogram) </pre>
<pre> <<block>> LiquefiedPetroleumGas : Fuel specificCarbonDioxide = "0.000000000250722"(dimension = EnergySpecificMass, unit = KilogramPerJoule) specificEnergy = "49300000"(dimension = MassSpecificEnergy, unit = JoulePerKilogram) </pre>

Figure 10: Abbreviated PEL for elements of the class Fuel

For the purposes of demonstration, only the carbon dioxide rate and specific energy of the Fuel is shown.

A similar library may exist for various Machines or Vehicles a user may pose, each containing values for fuels used, consumption rates, etc. These are defined only once by the user and seldom modified.

THE PARTICULAR INSTANCE MODEL

A user can create an instance of a ManufacturedPart that represents a manufactured part of interest. The user can then proceed to define all the Operations that are used to create that Part. A user does *not* need to define any of the value properties for the Part though. These will be calculated by the tool using only the information given in the Operations that define the ManufacturedPart.

Here is a critical feature of this method. The method requires only that a user define what is done to a Part during manufacturing, and what resources are consumed. The user does not need to calculate any numbers, the software will do the calculations internally using the plug-in ParaMagic. This may seem trivial, but the benefits become clear when dealing with large, complex elements that are being manufactured. In the case study discussed later, an aluminum Part model is created where only the Part itself, the actual Operations,

and which actors are used in those Operations are defined. All of the information about resource properties, actor properties, etc., has already been defined in the BCL or the PEL. The user has to create only a limited number of new instances and relationships, much of which can eventually be automated. This will be discussed in more detail in the Future Work section.

As an example of what a PIM would look like, given in Figure 11, shows the ManufacturedPart and Operation definition of an aluminum transmission case. Though the part that is being described may change, the final PIM will look very similar to what is seen in Figure 11.

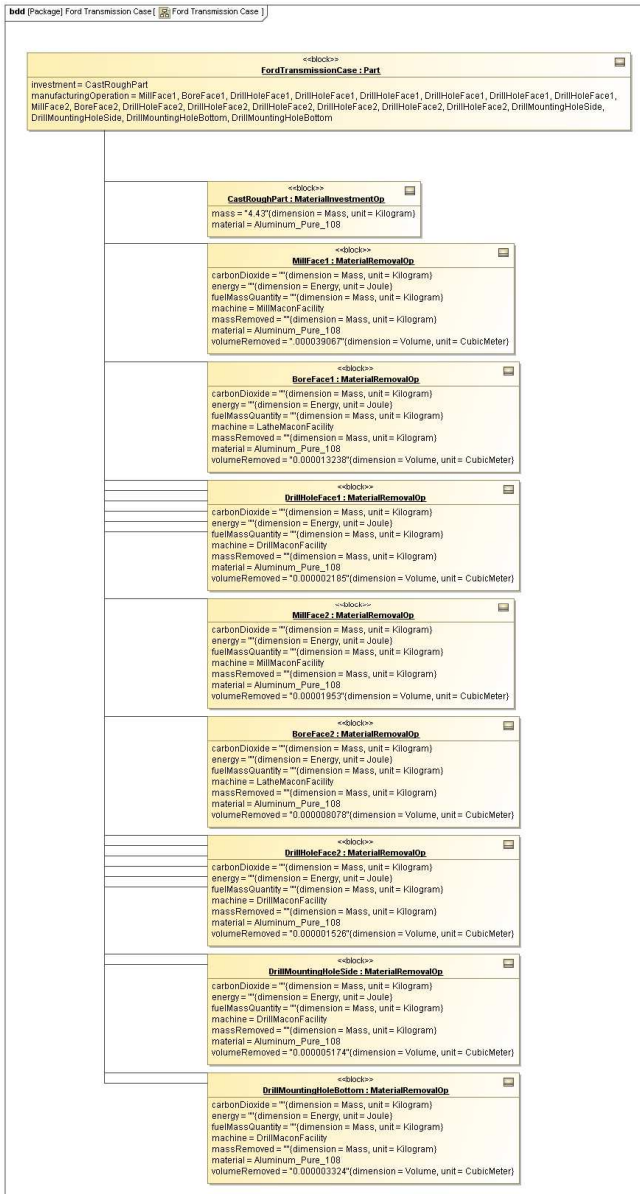


Figure 11: Example PIM showing a Part and its Operations

In the above figure, the ManufacturedPart is shown at the top. Below it is a list of Operations that it references, which in the software are linked to the Part with lines. Each line indicates a single Operation, so multiple lines to the same Operation denote multiplicity. Within each Operation, actors and materials are

assigned from the PEL. The only additional information entered here is the mass invested in the material investment operation, and the volume removed during the material removal operations. No transportation operations are shown here for the sake of simplicity.

SIMULATION AND EXECUTION OF THE MODEL

To run simulations and execute the model, the software tool uses a plug-in called ParaMagic, developed by InterCAX. ParaMagic parses through the model and extracts information about values and parametric relationships and imports them into Wolfram Mathematica. Mathematica then solves the parametric relationships with given values and ParaMagic takes the solutions and returns them to SysML where they can be updated to the model.

When solving, a browser window opens up. An example of the browser window can be seen in Figure 12.

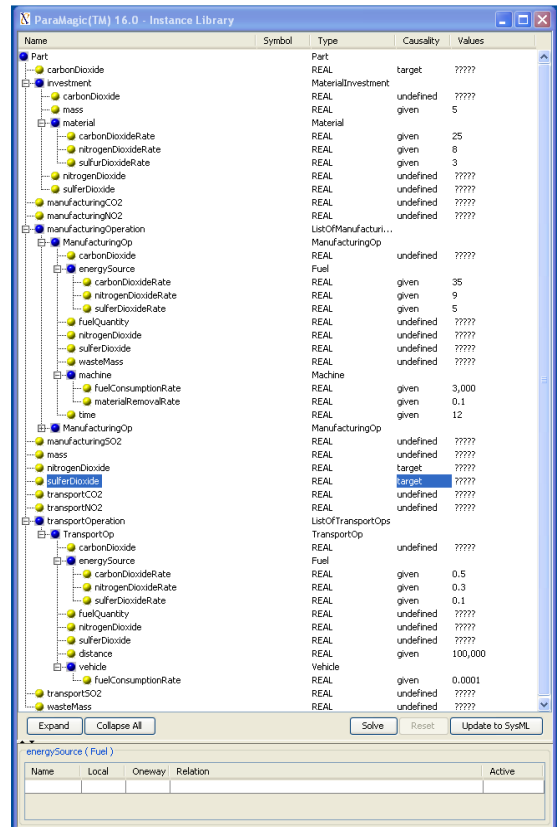


Figure 12: ParaMagic interface browser window

On the left hand side are the names corresponding to the values defined in the model. These names are identical to those in the model so they can be easily identified. Also shown is the containment of the values, so a user would be able to see in what elements values are contained, and in what higher level elements those lower level elements are contained. For each value, the type, causality, and numerical value are shown. The type corresponds to the value type assigned to each value property. These correspond to elements of the form given in Figure 4. Values that are numerically

defined in the model appear as “given” under the causality heading and their numerical values are shown. Values that are of interest and are the final goal of the calculation appear as “target.” The values that are not of particular interest and are not numerically given in the model appear as “undefined.”

Once the Solve button is hit, the browser window is updated to look as follows.

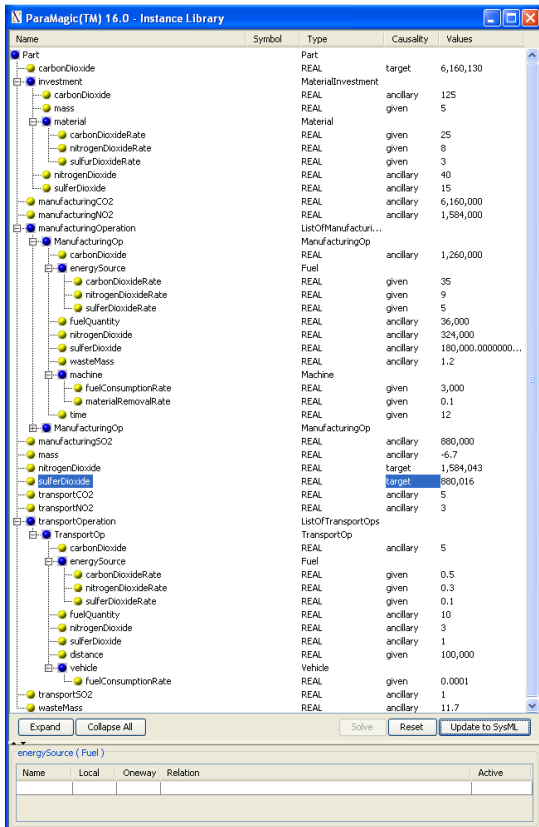


Figure 13: Solved browser for example system

The calculated numerical values for the target values appear in the browser. The causality of the undefined values changes once the system is solved. Undefined values that were not target values appear as “ancillary.” If the value was able to be computed, it is shown. Otherwise, the field is left blank. Ancillary values are solved “along the way” to the target values, and are not necessarily of primary interest.

At this point, the information can be updated back to the model by clicking Update to SysML. This should not be done unless the user wishes to change the model itself with the newly calculated values. The user always has the option to reset the system, which will return the browser to the values shown in Figure 12.

EXAMPLE USE CASE

SCENARIO AND OBJECTIVE

This section outlines the steps a hypothetical user would go through to implement the activity based assessment tool to the manufacturing process of a general aerospace part. The part is an aluminum part that is cast and machined to completion in a single

facility. The purpose here is to illustrate what a typical user may do to determine the manufacturing environmental impacts of a particular part.

In this example, the user has a physical description of the part of interest. This may be a CAD drawing, a scale mock-up, or the part itself. From this physical description, the user can acquire some information. Here, the user knows the material of the part and some geometric dimensions. In particular, the user can extract approximate geometric dimensions of features that are to be produced during the manufacturing process he wishes to model. These features can include diameter and depth of drilled holes, dimension of bored out segments, or area and depth of a machined cut out or cutaway. The user also knows what machines are present in the facility, as well as the process that is undergone to make this part.

The user wishes to calculate the total manufacturing related energy that is required to produce this part, as well as the total carbon dioxide emissions produced during the manufacturing process, given the above known information.

PARTICULAR ELEMENT LIBRARY – MATERIALS AND FUELS

The user already has available to him a library of materials and fuels. The elements in the library have predefined information that has been extracted from resource inventories. [citation]

Material properties, such as density and required cutting energy, are already defined from resource inventories. Additionally, a method by which the material was produced is already listed. The production method specified the approximate energy cost to refine a given mass of material. This information is available for a number of materials through various sources.

Fuels also have some information extracted from databases or inventories. In this case, the specific energy of a fuel and the carbon dioxide emissions per unit of energy produced for a fuel are given. For electrical power, the expected carbon dioxide emissions per unit of electrical energy is known from various environmental and government databases.

The above information has been predefined *before* the user begins modeling. However, the user is free to add fuels or materials and add or modify information as needed to most accurately describe the system of interest.

PARTICULAR ELEMENT LIBRARY - MACHINES

When the user first receives the software, he must create an inventory of available machines that can be used. These machines can be those that already exist in the facility, or can be ones not yet owned in the event the user wants to perform “what if” analysis. Each machine has a power source specified, chosen from the list of fuels in the PEL of fuels. Also, each machine has a maximum power output defined, which will help determine which machines are ultimately capable of performing a task.

The user would define this library *before* he begins modeling. Unlike the PEL for fuels and materials, the PEL for machines has to be created completely by the user. However, this library need only be created once and can be used for any subsequent model. Additional modifications to this library may be made as needed.

MANUFACTURED PART DESCRIPTION

The part is cast from pure aluminum using a natural gas furnace. The user has a CAD file with geometric specifications of features. A general geometric draft is given in Figure 14.

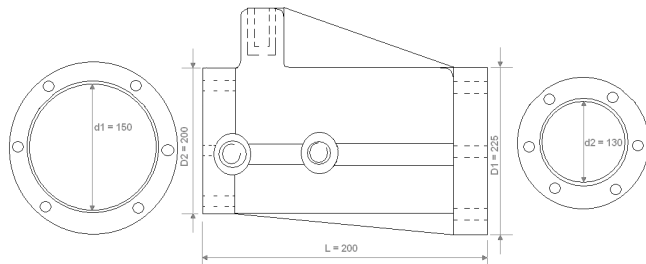


Figure 14: Hypothetical part CAD draft

The user would create a list of features that need to be created. This would come from the CAD file, or by looking at the manufacturing process plan for the part. For each of these features, the user would define the multiplicity and the approximate volume of material removed while making the feature. For the above part, the list of features is given in Table 1.

Table 1: Feature description for example part

FEATURE	VOL EACH (MM ³)	MULT	TOT VOL (MM ³)
Face 1	39 057	1	39 067
Face 1 Insert	13 238	1	13 238
Face 1 Hole	2 185	6	13 110
Face 2	19 534	1	19 534
Face 2 Insert	8 078	1	8 078
Face 2 Hole	1 526	6	9 156
Side Mounting Hole	5 174	2	10 348
Top Mounting Hole	3 324	2	6 648

This list can be generated multiple ways. The user can extract the data manually from a CAD software, the data from CAD software can be automatically generated, or the user can even measure and calculate the above information off of a physical model should the need arise.

Once the list of features is created, the user defines a list of operations that creates each of those features. This can take the form on “drill hole in face 1” or any other descriptive title. The operations that describe the production of the features in Table 1 are all material removal operations. The user must add an additional material investment operation that represents the initial casting of the part. For the material investment operation, the user plugs in a machine that will perform the casting, such as a coal furnace or an electric induction furnace in a sand casting process, and then

specify the approximate mass and type of material he wishes to invest. For each of the material removal operations, the user specified the volume of material removed, the material being removed, and the machine that is performing the operation. Notice that the machine and the material are taken from the PEL, so the user just needs to reference these elements, not recreate them.

This would conclude the modeling aspect of the part. Notice the information that the user has supplied: material used, machines used with power source specified, mass of invested material, and volume of material removed by operation. The user does not know anything yet about the total amount of energy required to perform these operations, nor does he know what the carbon dioxide emissions could be. The tool would perform these calculations for the user. These calculations can be done by hand, but would get too complicated to be reliable once the system grows. The sample part given above already includes 8 different features that will take 21 individual operations to produce. The computer aided method takes the burden of manually calculating data off of the user. Also, the object oriented approach allows for easier storage and manipulation of data within the software. By using an activity based approach where the user defines a part in terms of the operations (activities) that are performed on it, the user can generate a model based on available information. A user in a factory would know, for instance, the available machines or processes used in creating parts, but would not necessarily know the carbon dioxide emissions produced during those processes. This is the purpose of the method: to define a part in terms that the user can more easily define, such as operations, and calculate environmental impact information using previously tabulated inventory data and user input specifications.

For the user to actually calculate the values of carbon dioxide and energy required, he would execute the model. This tool, created in SysML, would execute the model using the plug-in ParaMagic and the mathematical tool Wolfram Mathematica. The user would browse the data that he has input in a browser tree similar to Figure 1. He would specify target information under the heading ManufacturedPart. In this case, ManufacturedPart was defined in Figure 8 with values of manufacturingCarbonDioxide and manufacturingEnergyConsumed (amongst other values). These would be the target values the user would look for. The user clicks Solve in the browser window, and the results would be computed and displayed in the browser window. All the values that were able to be calculated are calculated and those that do not have enough information to be calculated are left blank. Should there not be enough information to calculate target values, the tool will register an error message, and the user would know that there is insufficient information in the model.

The results would vary greatly depending on the material or the energy source the user chose, and even the machine the user chose to use. Rather than having to re-compute the carbon dioxide emissions or the energy consumed by hand, the user simply replaces the

slots for material and energy source with a different material or energy source and possibly the machines that were used. Then the user would open another browser window and would hit Solve again. This would allow the user to do "what if" analysis fairly easily.

FUTURE WORK

FUTURE WORK ON THE SOFTWARE TOOL

The tool needs to be refined so that the definitions in the Base Classifier Library more accurately represent real life elements and relationships. Furthermore, the definition for the operations needs to be modified, again to more accurately be able to describe real life systems. Additional diagrams, such as activity or sequence diagrams, will be created to add another level of detail. The existing model will also be fleshed out with additional sustainability and environmental impact indicators and values. The model, as is, only tracks a handful of emissions and energy consumption. The list of indicators and values that are tracked will be expanded so that the tool can best describe the environmental impacts of a manufacturing process.

At this stage, these additional diagrams will serve only as descriptive representations of the system. Descriptions beyond what has been shown in this article has limited to no executable capability.

SysML is a systems realization software. It can represent and store a large amount of information, but SysML itself does not have executable capabilities. Simulation or execution of a model defined in SysML requires additional plug-ins. Currently, ParaMagic has the capability to solve information in instance diagrams and parametric diagrams. ParaMagic has, in development, features that allow it to interface and export information to Microsoft Excel and Matlab.

Research departments at Georgia Institute of Technology are working on additional plug-ins to interface the SysML model with Microsoft Excel, Factory CAD software, E-Plan software, and other tools that can help add simulation capabilities to the model. These tools are still in development though.

FUTURE WORK ON THE METHOD

The method needs to be validated. Activity based costing has been well established, but the application to sustainable manufacturing system analysis needs to be tested. Applying the method to a real world example and comparing to experimentally gathered data can help validate the method and the tool.

CONCLUSION

It is clear that minimizing environmental impact is becoming more and more important, making the need for a sustainability assessment method necessary. For certain industries, such as the manufacturing industry, such a method is greatly lacking. To implement this method, a tool will be used. The activity based approach that uses computer aided, object oriented software shows that it can be used to define a manufacturing

process for a manufactured part. The method defines a manufactured part in terms of the operations that are performed on it and the resources it consumes. Such an approach is beneficial to the manufacturing industry because it capitalizes on using information that is more readily available to manufacturers, such as product dimensions and manufacturing process planning. The approach uses the known information from the manufacturer and combines it with environmental inventory data to create a sustainability assessment tool that a manufacturer can implement. The tool has shown that it is both flexible and expandable without changing the underlying method behind the modeling. Overall, this method shows potential for providing the manufacturing industry a greatly needed tool in determining their environmental impact.

ACKNOWLEDGMENTS

We would like to acknowledge Lisa Phelps along with the Boeing Company and Boeing Phantom Works, specifically those involved in the St. Louis, MO and Macon, GA facilities for providing us with funding and critical feedback with the development of this method and tool. We would also like to acknowledge Dr. Russell Peak and Dr. Manas Bajaj and the people at the Manufacturing Research Center at Georgia Tech and the people at InterCAX for their work and help with ParaMagic. We would also like to acknowledge Dr. Leon McGinnis and Dr. Chris Paredis of Georgia Tech for their work on and with SysML and other supporting software, as well as for their introducing and instructing us on SysML and supporting software.

REFERENCES

1. (2008). "Official Energy Statistics from the US Government." from <http://www.eia.doe.gov/oiaf/1605/coefficients.html#note1>.
2. (2009). "Coal Statistics." 2009, from <http://www.worldcoal.org/pages/content/index.asp?PageID=187>.
3. (2009). "Energy Density." 2009, from http://en.wikipedia.org/wiki/Energy_density.
4. Bennet, M. and P. James (1999). Sustainable Measures. Sheffield, UK, Greenfield Publishing Limited.
5. Cattanach, R. E., J. M. Holdreith, et al. (1995). The Handbook of Environmentally Conscious Manufacturing. multiple cities, Richard D. Irwin, Inc.
6. DeGarmo, E. P., J. T. Black, et al. (1997). Materials and Processes in Manufacturing. Upper Saddle River, NJ, Prentice-Hall Inc.

7. Emblemvag, J. and B. Bras (2001). Activity-Based Cost and Environmental Management. Boston, Dodrecht, London, Kluwer Academic Publishers.
8. Friedenthal, S., A. Moore, et al. (2008). A Practical Guide to SysML, Morgan Kaufmann Publishers (an imprint of Elsevier).
9. Gabbar, H. A. (2009). "Engineering Design of Green Hybrid Energy Production and Supply Chains." Environmental Modelling & Software **24**(3): 423-435.
10. Gonzalez-Benito, J. (2008). "The Effect of Manufacturing Pro-Activity on Environmental Management: An Exploratory Analysis." International Journal of Production Research **46**(24): 7017-7038.
11. Govetto, S. (2008). Determinin the Environmental Impact of Remanufacturing End-of-Life Strategies Compared to Recycling and Disposal Strategies. Mechanical Engineering. Atlanta, Georigia Institute of Technology. **Masters of Science: 278**.
12. Hsueh, N.-L., J.-Y. Kuo, et al. (2007). "Object-Oriented Design: A Goal-Driven and Pattern-Based Approach." Software & Systems Modeling **8**(1): 67-84.
13. Kalpakjian, S. and S. R. Schmid (2001). Manufacturing Engineering and Technology. Upper Saddle River, NJ, Prentice-Hall, Inc.
14. Pahl, G. and W. Beitz (1996). Engineering Design: A Systematic Approach, Springer-Verlag London Limited.
15. Pineda-Henson, R. and A. B. Culaba (2004). "A Diagnostic Model for Greed Productivity Assessment of Manufacturign Processes." International Journal of Life Cycle Assessment **9**(6): 379-386.
16. Reap, J., F. Roman, et al. (2007). "A Survey of Unresolved Problems in Live Cycle Assessment (Part 1)." International Journal of Life Cycle Assessment **13**(4): 290-300.
17. Reap, J., F. Roman, et al. (2008). "A Survey of Unresolved Problems in Life Cycle Assessment (Part 2)." International Journal of Life Cycle Assessment **13**(5): 374-388.
18. Rivera, J. and J. Sutherland (2008). "Application of Life Cycle Assessment Tools to Sustainable Product Design and Manufacturing." International Journal of Innovative Computing Information and Control **4**(3): 577-591.
19. Timings, R. L. (1998). Manufacturing Technology, Addison Wesley Longman Limited.
20. Tlusty, G. J. (2000). Manufacturing Processes and Equipment. Upper Saddle River, NJ, Prentice-Hall, Inc.
21. Xiaoming, Z. and D. Pingan (2009). "A Model-Based Approach to Assembly Sequence Planning." International Journal of Advanced Manufacturing Technology **39**(9/10): 983-994.
22. Xu, X., J. L.-Q. Chen, et al. (2006). "Framework of Product Lifecycle Costing System." Journal of Computing and Information Science in Engineering **6**(1): 69-77.
23. Yang, D. and H. Wu (2009). "A UML-Based Approach for the Development of Shop Floor Control Systems." International Journal of Production Research **47**(6): 160.