

A SUSTAINABILITY TOOLKIT FOR SIMULATION: RECENT DEVELOPMENTS AND FUTURE CAPABILITIES

Xi Zhou
Michael E. Kuhl

Rochester Institute of Technology
81 Lomb Memorial Drive
Rochester, NY 14623 USA

ABSTRACT

The use of simulation to study complex systems having both productivity and sustainability related performance measures seems to be growing at a rapid pace. This paper describes the recent developments and discusses potential future capabilities of a sustainability toolkit for simulation. The intent of the toolkit is to make sustainability related performance measures as easy to model and collect as traditional productivity based performance measures. The current toolkit focuses on the environmental aspects of sustainability, however, the goal of the toolkit is to also include the social and economic aspects of sustainability in the near future.

1 INTRODUCTION

The scope of simulation modeling and analysis has changed over the last several years from focusing primarily on productivity-related system performance measures to include performance measures related to sustainability. Sustainability issues including energy consumption, emissions and environmental impact, have become increasingly important factors in many business decisions. Thus there has arisen a need for simulation tools to efficiently enable the collection and reporting of sustainability related measures. We have undertaken an effort to develop a sustainability toolkit for simulation in which we supplement traditional constructs and performance measures with a sustainability toolkit that contains a flexible framework to enable the modeling and analysis of sustainability factors and performance measures.

Sustainability is often seen as having three major aspects or components – a social aspect, an economic aspect, and an environmental aspect – which are interdependent (see Figure 1). The long term plan for the sustainability simulation toolkit is to provide toolset that takes an integrative systems approach to modeling and analyzing problems including all three aspects of sustainability.

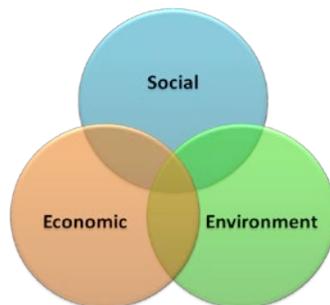


Figure 1: Three major aspects of sustainability – social, economic, and environment

2 SUMMARY OF THE TOOLKIT METHODOLOGY

The current sustainability toolkit focuses on the process-oriented aspects of sustainability. The general methodology for developing the process-oriented, environmental modules of the sustainability simulation toolkit were introduced and detailed in Kuhl and Zhou (2009) involves the following steps:

- Identify sustainability factors and environmental performance measures for systems of interest;
- Establish simulation state variables to represent the sustainability factors and performance measures;
- Identify the events in the system that cause changes to the state variables, and translate these system events to simulation events and the associated event triggers;
- Establish the mathematical, statistical, and logical relationships among the state variables to update and change the values of the state variables as events occur in the system over time;
- Implement state variables, relationships, and events into a robust, flexible simulation toolkit modules; and
- Perform verification, validation, and testing of the simulation toolkit modules.

Since we are utilizing discrete-event simulation, the changes to the state-variables occur upon the instantaneous occurrence of events. For items such as emissions which may occur continuously over an interval of time, we assume (at least initially) that emission quantity between event occurrences can be represented as a function of time, a probability distribution, or a stochastic process. In the future, it may be necessary to represent some factors as continuous variables with respect to time and utilize difference and/or differential equations to accurately represent some systems. Upon determining the mathematical, statistical, and logical relationships among state-variables, they are implemented into the sustainability toolkit consisting of a flexible framework of modules which are developed for integration into off-the-shelf and customized simulation software. Initially, we are implementing the modules in the ARENA simulation software. Upon gaining these capabilities our intention is to implement the modules in a general programming language such as C++ and provide “hooks” on the toolkit modules that will allow for easy adaptation to off-the-shelf simulation languages.

3 CURRENT DEVELOPMENT AND IMPLEMENTATION OF TOOLKIT MODULES

The development of the sustainability toolkit has resulted in the implementation of modules for modeling and analysis of three important and commonly encountered system components including transportation and logistics, industrial processes, and material handling. In addition, a set of flexible modules have been developed for the collection and reporting of general performance measures. Figure 2 depicts the ARENA template that has been created using the template development tools provided in ARENA. The modules created in this template can be used in conjunction with the other templates and modules provided with ARENA. In the following sections, we describe the development and functionality of the modules and provide examples to illustrate their use.

The set of modules developed for the modeling and analysis of transportation and logistics systems, industrial processes, and material handling systems are discussed in detail in Zhou and Kuhl (2010). For instance, the *Transportation and Logistics* modules are used to model and collect emissions generation from running trucks or other vehicles in transportation systems. *Material Handling* modules are used to model warehouse operations where transporters such as forklifts are used. *Process with Emissions* module is designed to collect the environmental measures from industrial and manufacturing processes. The system components and sustainability performance measures that are modeled explicitly for these systems are summarized in Table 1. In addition to these performance measures, the modules are constructed to enable the use to modify predefined measures of performance and to add additional performance measures. In addition, Zhou and Kuhl (2010) provide illustrative examples that demonstrate the use and flexibility of these modules.

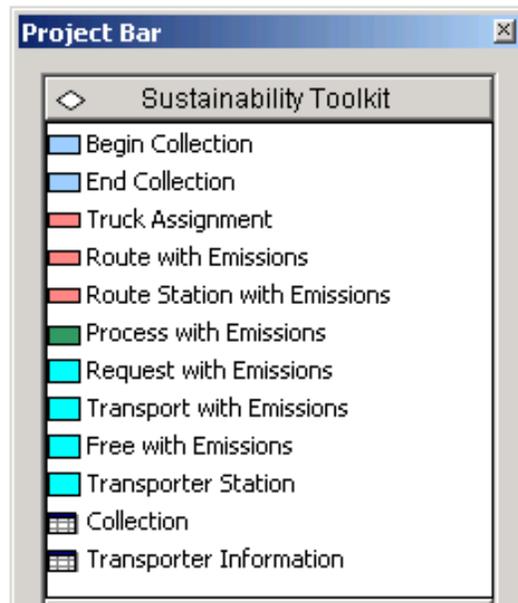


Figure 2: Current sustainability toolkit implementation

Table 1: Environmental Performance Measures collected by Sustainability Toolkit

Systems and System Components	Environmental Performance Measures
Transportation and Logistics Systems	Fuel Consumption CO, CO ₂ , NO _x , THC Generation
Industrial and Manufacturing Processes	
<ul style="list-style-type: none"> • Industrial Coating Process 	Paint and Solvent Consumption VOC Generation
<ul style="list-style-type: none"> • Injection Molding Process 	Energy Consumption CO ₂ , SO ₂ , NO _x , CH ₄ Generation
<ul style="list-style-type: none"> • Plastics Processing 	Energy Consumption Water Usage CO ₂ , SO _x , NO _x Generation
Material Handling Systems	HC, CO, NO _x , PM Generation

3.1 General Collection Modules Development

The sustainability toolkit modules discussed in the previous section are used to model particular application areas. In developing these application specific modules, we gained experience and demonstrated that the application specific modules can be very useful. Since there are many other systems for which the application specific modules cannot be used, before developing additional application specific modules, we decided to develop a generalized version of a performance measure collection and reporting module. Figure 3 depicts the general collection modules developed including *Begin Collection* module, *End Collection* module, and *Collection* data module. Simply, the *Collection* data module defines a performance measure (collection) with information such as collection type and expression relationship. *Begin Collec-*

tion module initializes a specified collection and *End Collection* module ends a specified collection and stores corresponding statistics.



Figure 3: General collection modules

The common pattern used in collecting the application specific performance measures in the discrete-event simulation framework is summarized in Figure 4. The three basic steps include first assigning an initial value to the performance value and indicating that the collection of the performance measure should begin. Second, the entity proceeds through the set of logic where operations/processes are performed (perhaps over some time duration). Finally, when the operation has ended, the performance measure of interest is calculated.



Figure 4: Three steps in general to collect simulation system performance measures

More specifically, in discrete event simulation *operations* are often defined as the steps carried out by the simulation entities. The entity flows through the simulation system and carries out the operations. Two sustainability-related simulation flows operations, the *Beginning of Collection* and the *End of Collection* operations, are defined. Taking the emissions generation for instance, when an entity carries out the *Beginning of Collection* operation, it means that the entity starts to generate emissions (e.g. the truck starts to route to the next destination), or some other resource in the system starts to generate emission (e.g. the parts start to request a forklift to come). The *Beginning of Collection* operation is controlled and triggered by the simulation entity. Initial values of the parameters for calculating the environmental measures are assigned. After the entity carries out the *Beginning of Collection* operation, it moves to other operations dependant on the systems being modeled, during which period of time, the emissions are generated continuously within the system. As soon as the entity, or other resource in the system, stops generating the emissions, the *End of Collection* operation needs to be executed. In the *End of Collection* module, the environmental measures are calculated using the initial values assigned in the *Beginning of Collection* module along with the mathematical relationship defined by the user in the *Collection Data Module*.

3.1.1 The Collection Data Module

The *Collection* data module is used to specify the user-defined collection with mathematical expression of how to calculate the collection (see Figure 5).

Collection - Sustainability Toolkit							
	Collection Name	Collection Type	Rows	Member Value	Delta	Delta Type	Accumulator Expression
1	CO2 generation	Entity Accumulator	8	8 rows	<input checked="" type="checkbox"/>	TNOW	Delta * Truck HP * Truck Ideling * EmissionCoe(1)
2	NOx generation	Entity Accumulator	8	8 rows	<input checked="" type="checkbox"/>	TNOW	Delta * Truck HP * Truck Ideling * EmissionCoe(2)
3	System CO2 generation	System Accumulator		0 rows	<input type="checkbox"/>	TNOW	CO2 generated by truck
4	System NOx generation	System Accumulator		0 rows	<input type="checkbox"/>	TNOW	NOx generated by truck

Figure 5: General Collection Modules: *Collection* data module

In the first field, a name is given to the performance measure being collected (Collection Name). The data module allows two types of performance measure that are entity specific or system performance measures. We refer to these as an entity accumulator and a system accumulator, respectively. The collection name is unique representing collection i . An entity accumulator collects the specified collection based on entity. Therefore, in order to know what group of entities is collection the statistics, the operand ‘Rows’ and ‘Member Values’ are used to define the collecting group (G_i). In the simulation model, only the entities which match the member value will perform the calculation and return the collection by entity. System accumulator collects the defined collection for the entire simulation system. It is found that the difference value between two parameters is often used in the calculation expression. Therefore, we create a convenient for the user to input the difference ‘delta’ ($Delta_i$). Three commonly used delta type are created as listed in Table 2. The ‘TNOW’ and ‘Attribute’ types are most commonly used deltas where ‘TNOW’ represents the time interval and ‘Attribute’ means a difference based on some entity property. The notations and definitions of the system parameters are defined in Table 3.

Table 2: Delta Type

Delta Type	Delta Definition
TNOW	time interval
Attribute	difference based on a particular entity attribute
Expression	difference based on a particular expression

Table 3: Notations of Parameters used in the General Collection modules

Notations	Definitions
i	User-defined collection i
G_i	Group of entities that user specifies to collect i (can be an empty group)
$Delta_i$	A predefined variable used in user-defined collections which represents the changes in a system parameter
$Delta_{Start_i}$	Initial value of $Delta$
$Delta_{End_i}$	End value of $Delta$
Var_i	State variable representing collection i
Exp_i	User-defined expression to update Var_i

3.1.2 Begin Collection Module

The *Begin Collection* module is where the molder specifies which collection, as defined in the *Collection data module*, to start collecting (see Figure 6). If any delta is used in the calculation of the collection, the *Begin Collection* module assigns the initial values of the delta used. Figure 7 shows the logic flow within the *Begin Collection* module.

If delta is involved in the calculation, the *Begin Collection* module assigns the initial or starting value of the delta depending on the delta type. A time interval delta has an initial value of the current simulation time. An entity attribute-based delta has an initial value of the current value of the specified attribute while an expression-based delta has an initial value of the current expression value.

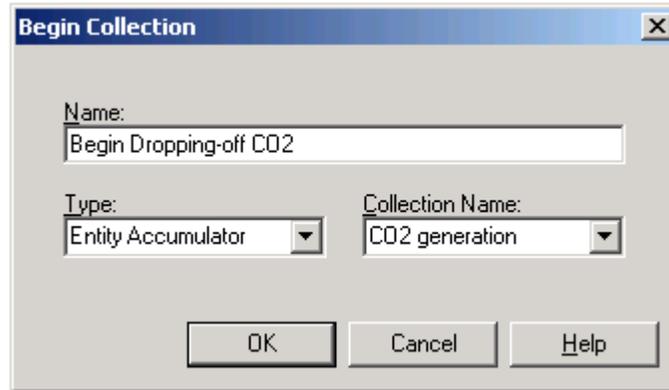


Figure 6: General Collection Modules: *Begin Collection* module

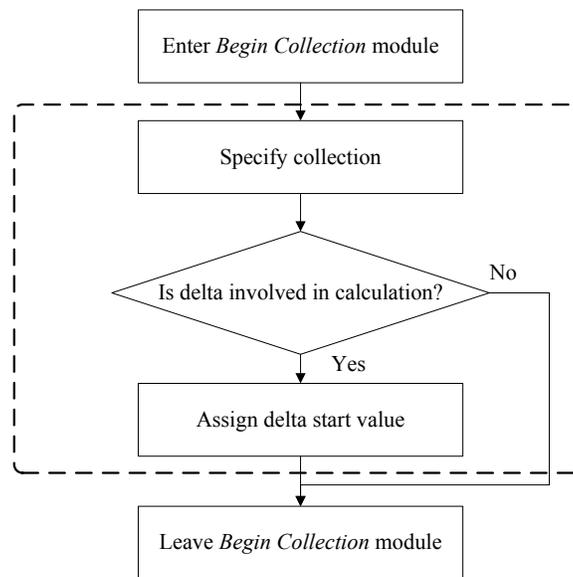


Figure 7: Flowchart of the logic for the *Begin Collection* Module

3.1.3 End Collection Module

The *End Collection* module ends a predefined collection by calculating and storing the corresponding environmental performance measure (see Figure 8).

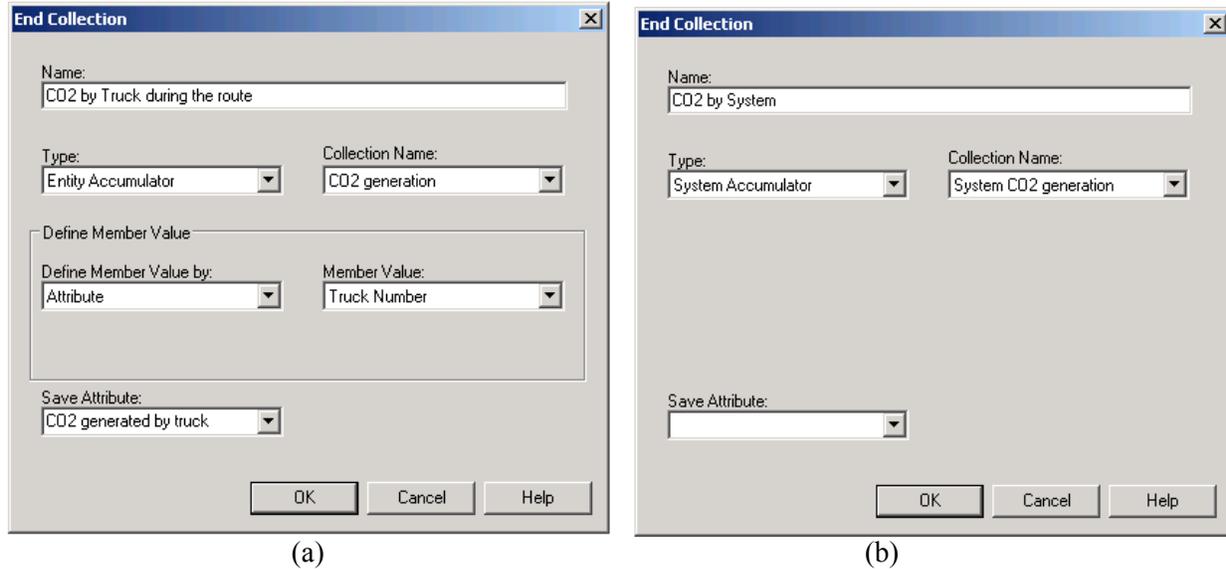
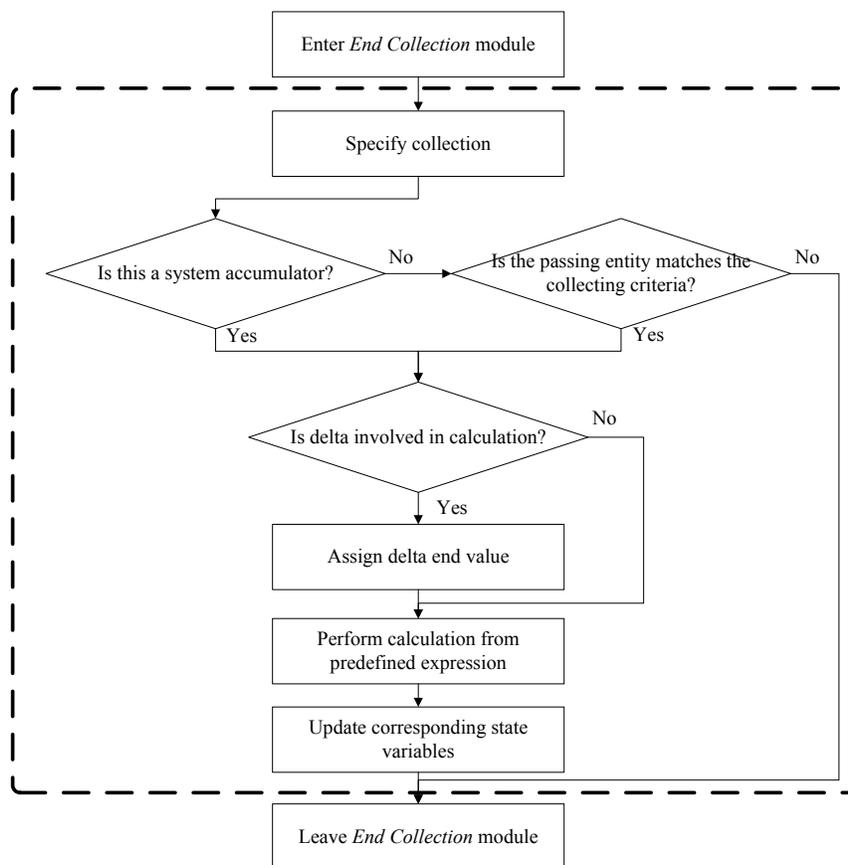


Figure 8: General Collection Modules: *End Collection* module, (a) Entity Accumulator, and (b) System Accumulator

The module first determines which collection to be ended. If the collection is an entity accumulator, the module checks whether the collection is calculated by this currently passing through entity. The collecting criteria can be evaluated either by entity attribute or by an expression. If the determination returns to be true, the *End Collection* module calculates the performance measure using the expression defined in the *Collection* data module. The *End Collection* module also updates the entity accumulator by the passing entity. If the passing entity is not collecting performance measures, in other words, the passing entity is not in the group of the specified entity accumulator, it flows through the *End Collection* module without performing any calculations. System accumulator updates the system performance measures by the amount of value evaluated from the expression defined in the *Collection* data module. Last, the user can choose to save the evaluated expression value as an attribute of the passing entity. A flow chart of the logic in the *End Collection* module is shown in Figure 9.

3.2 Summary of the General Collection Modules

The general collection modules generalize the sustainability toolkit in such a way that the modules are able to be used to any systems of interest. The modeler only needs to specify the collection, including the collection type and the formulation of updating the collection, for one time in the simulation model. When constructing the simulation model, the user only needs to determine what is the beginning and the end of an environmental collection, and applies the general collection modules. The beauty of the modules is that least modules (three modules) can provide a high level flexibility of simulation modeling to collect the user-defined environmental performance measures in an easy way. The modeler has full control of what are collecting and how to collect and at the same time has an efficient input manner and a ready-to-read automate output.

Figure 9: Flowchart of the logic for the *End Collection* Module

4 RECOMMENDATION FOR FUTURE RESEARCH

The current sustainability toolkit demonstrates the usability of the toolkit development and implementation methodology. The toolkit implementation shows the abilities provided by the sustainability toolkit. The following list provides some recommendations for future research that may further expand the usage of the sustainability toolkit:

Obtain databases for evaluating the environmental impact using simulation. The aspect requires that the data and mathematical relationships to evaluate the environmental impact can be obtained from new research studies. Databases can be constructed and specified within the sustainability toolkit modules in order to identify and inform the end user of which database is applied and used.

Establish modules for other applications of environmental impact. This aspect includes two components. One component is to complete more options to the currently developed sustainability toolkit. For instance, the plastics processing component in the process with emissions module can be further designed to include more plastic options. The other component of this recommendation is to develop sustainability toolkit modules for other systems of interests. For instance, ocean fishing systems have the environmental impact of vessel fuel usage, and emissions generated from burning the fuel. Sustainability toolkit modules can be designed to represent the ocean fishing system of its environmental impact.

Extend sustainability toolkit to other aspects of sustainability. This research's scope mainly focuses on the environment aspect of sustainability. However, the development methodology could potentially be applied to other two aspects of sustainability, namely the economic aspect and the social aspect, to help in policy making. Taking the ocean fishing system for instance, potential sustainability impact of ocean fishing may include the overfishing. Sustainability toolkit could be applied to ocean fishing system to evalu-

ate the overfishing performance measure and hence assist in policy making such as determining the length of open season for fishing. This application falls into the economic aspect of sustainability therefore the scope of the toolkit can be expended.

Implement sustainability toolkit in general programming language SysML: SysML is an extension of UML that are used specifically to support systems engineering. It is a system modeling language used for creating descriptions of systems. Thus, implementation of the sustainability toolkit in SysML addresses the common problem of model validation. Furthermore, since it is a general modeling language, writing sustainability toolkit in SysML has the benefit of translating the toolkit to various simulation packages. Therefore, implementing the sustainability toolkit in SysML may be a good approach for future research and study.

REFERENCES

- Kuhl, M.E. and Zhou, X. 2009. "Sustainability Toolkit for Simulation-based Logistics Decisions." In *Proceedings of the 2009 Winter Simulation Conference*, edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls, 1466-1473. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.
- Zhou, X. and Kuhl, M.E. 2010. "Design and Development of a Simulation-based Sustainability Toolkit." In *Proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hagan, and E. Yücesan, 1601-1612. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

AUTHOR BIOGRAPHIES

XI ZHOU is a graduate student pursuing a Master's degree in Industrial and Systems Engineering at Rochester Institute of Technology. His research interests are in field of systems modeling and analysis. His e-mail address is xi.zhou@mail.rit.edu.

MICHAEL E. KUHL is a Professor in the Industrial and Systems Engineering Department at Rochester Institute of Technology. He has a Ph.D. in Industrial Engineering from North Carolina State University (1997). His research interests include simulation modeling methodologies with application to cyber security, healthcare, and sustainability. He served as Proceedings Editor for the 2005 Winter Simulation Conference. He is a member of the INFORMS Simulation Society, IIE and ASEE. His e-mail address is Michael.Kuhl@rit.edu and his Web address is people.rit.edu/mekeie.