Rochester Institute of Technology
RIT Digital Institutional Repository

Theses

8-1-2011

Design and development of a sustainability toolkit for simulation modeling and analysis

Xi Zhou

Follow this and additional works at: https://repository.rit.edu/theses

Recommended Citation
Zhou, Xi, "Design and development of a sustainability toolkit for simulation modeling and analysis" (2011). Thesis. Rochester Institute of Technology. Accessed from

This Thesis is brought to you for free and open access by the RIT Libraries. For more information, please contact repository@rit.edu.

Rochester Institute of Technology

DESIGN AND DEVELOPMENT OF A SUSTAINABILITY TOOLKIT FOR SIMULATION MODELING AND ANALYSIS

A Thesis

Submitted in partial fulfillment of the

requirements for the degree of

Master of Science in Industrial Engineering

in the

Department of Industrial & Systems Engineering

Kate Gleason College of Engineering

by

Xi Zhou

August, 2011

DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING

KATE GLEASON COLLEGE OF ENGINEERING

ROCHESTER INSTITUTE OF TECHNOLOGY

ROCHESTER, NEW YORK

CERTIFICATE OF APPROVAL

M.S. DEGREE THESIS

The M.S. Degree Thesis of Xi Zhou

has been examined and approved by the

thesis committee as satisfactory for the

thesis requirement for the

Master of Science degree

Approved by:

____

Dr. Michael E. Kuhl, Thesis Advisor

____

Dr. Andres Carrano

____

Dr. Brian Thorn

ABSTRACT

A simulation-based sustainability toolkit is developed in this research, in order to efficiently provide the capability of modeling and evaluating the sustainability related performance measures in systems of interest. As sustainability related issues, such as energy consumption, emissions, and waste generation are becoming a more integrated part of operational and long-term planning decisions, simulation modeling is needed to analyze the system performance not only from the traditional system performance measures such as productivity and efficiency, but also taking into account the environmental impacts within the studied system in order to aid in decision-making. This research introduces the concept and general methodology to develop such a sustainability toolkit for simulation and provides a prototype implementation of the toolkit using commercially available discrete event simulation software. In particular, toolkit modules have been developed for modeling and evaluating the sustainability aspects of transportation and logistics systems, industrial and manufacturing processes, and warehouse material handling systems. Furthermore, general sustainability toolkit modules are constructed to model other systems of interest. The toolkit contains a flexible framework which enables the simulation modeling and analysis of the sustainability related performance measures as easily as traditional system performance measures. The toolkit will enable the users to efficiently simulate complex systems taking into account system sustainability in an integrated decision-making process.

TABLE OF CONTENTS

1 INTRODUCTION

Sustainability related issues, such as energy consumption, emissions and overall environmental impact, have become increasingly important in business decisions. However, traditional decision support tools such as simulation have primarily focused on the operational aspects of industrial and service systems such as efficiency, productivity, and cost with little attention given to the efficient analysis of environmental impact. A sustainability toolkit for simulation modeling and analysis is needed to supplement the traditional system performance measures with sustainability related system performance measures. This toolkit is intended to contain a flexible framework to enable the modeling and analysis of sustainability-related factors and performance measures. The toolkit should also be designed for ease of implementation, so decision-makers can have sustainability performance measures as readily available as traditional performance measures when making decisions.

Sustainability is commonly viewed as having three dimensions: the social dimension, the economic dimension, and the environmental dimension (United Nations General Assembly, 2005). The social dimension mainly focuses on the human health and safety issues and also the human relationship to nature. The economic dimension is a subfield of economics dealing with the economic effects of sustainability issues. The environmental dimension studies how nature and the environment (e.g. the ecosystem) changes and affects humans (see Figure 1-1).

Industrial and service businesses have grown extremely fast during the past century, and productivity has often taken a higher priority in decision making than the environmental impacts of a manufacturing or service system. As a cost, world energy consumption and waste generation has increased dramatically. According to Energy Information Administration (2008), total worldwide energy consumption was 472 quadrillion Btu in 2006. It is also estimated that the

global demand for energy will increase by as much as 50 percent by 2025 (DOE, 2006). Waste, including municipal solid waste, commercial waste and hazardous waste, has been generated and emitted to the environment. U.S. Environmental Protection Agency reports that in 2007, there was 254 million tons of municipal solid waste that was generated in United States, which is 4.62 pounds per person per day (EPA, 2008). The Energy Information Administration (2007) reports that total U.S. carbon dioxide emissions in 2007 were 6,022 million metric tons, which increases by 17% over that in 1990. Total U.S. methane emissions in 2007 were 700 million metric tons. Both gases, known as greenhouse gases, are largely produced by industries such as energy generation industry and manufacturing industry. Parallel to the increasing trends in energy consumption and waste generation, the cost associated with the energy consumption and waste disposal is increasing as well due to the limitation of overall available energy and land.

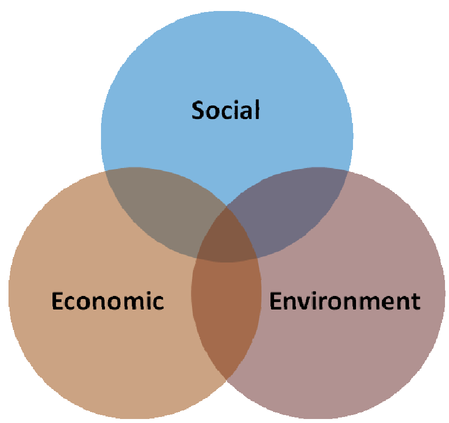


Figure 1 - 1: Three Dimensions of Sustainability

As a result, sustainability related issues have become increasingly important and represent a necessary support component in decision-making. When evaluating a system configuration, decision-makers should be able to examine the system not only from the traditional performance measures such as productivity, efficiency and cost, but also take into account of the sustainability related aspects as well. For instance, long-term decisions such as the

material selection in manufacturing systems or the fuel alternatives determination in transportation systems are especially affected by the corresponding environmental impacts. The efficient and effective analysis of sustainability related system performance measures is becoming more important to decision-makers.

2 PROBLEM STATEMENT

Simulation modeling is a tool which is used widely to model complex systems and evaluate system performance, in order to help with decision making. Simulation has been proved to be an efficient method in modeling industry and service systems where the experimentation with real system configurations is costly (Law, 2007). Simulation modeling assists the decision maker to analyze the system configuration without actually running the real system. Traditional system performance measures in simulation consist of measures such as productivity, efficiency, resource utilization, and cost. The current off-the-shelf simulation packages are so well designed that most of the traditional performance statistics can be shown to the user as a part of the default output report. With the trend of increasing energy cost and invention of environment friendly resources, businesses are taking into account sustainability impact when analyzing system configurations.

Currently there exist few simulation packages which are able to evaluate the sustainability issues as a built-in integrated part of the simulation software. In order to model a system with sustainability performance measures, more modeling effort and time is required than modeling of traditional performance measures. The existing simulation tools have primarily focused on the operational aspects of industrial and services systems with little attention given to the sustainability impacts. This research intends to supplement the traditional performance measures that are provided in current simulation packages with the system performance measures that are environment focused. To address this issue, there is a need to develop a simulation-based sustainability toolkit. The sustainability toolkit should have the capability of evaluating the environmental impacts and collecting sustainability performance measures. Thus, the key objectives of this research are to:

- *Design a methodology for creating generalized modules for modeling and collecting sustainability measures*: The development methodology should cover the aspects including selecting a study system, defining the sustainability performance measure, defining the simulation state variable, determining the simulation events, and formulating the mathematical and logical relationship in order to perform the collection of the sustainability statistics.
- *Identify systems and system components where the application of the sustainability toolkit would have significant impact in decision-making*: To develop such a sustainability toolkit, the application areas of the toolkit are researched and identified. The toolkit is developed and used for the specified systems of interest. The sustainability issues within a particular system should be evaluated to determine if there is a need to develop such a sustainability toolkit for the system.
- *Implement and integrate the sustainability toolkit to off-the-shelf commercial simulation software and produce a generalization of the toolkit*: A target simulation package is selected to implement the development of the sustainability toolkit. The objective of developing the toolkit also includes providing the users with an easy-to-use implementation of a flexible structure that allows for user-specified changes and updates. Therefore, a generic sustainability toolkit needs to be produced which is application area independent.
- *Conduct verification, validation and testing of the sustainability toolkit*: In order to provide credibility and trust to the user to use the sustainability toolkit, the toolkit needs to be verified, validated and tested to have the desired usage ability, which is to collect and analyze the sustainability-related system performance measures, and to report the

measures as readily as other traditional system performance measures. The assumptions made in developing the toolkit need to be documented and verified.

With the development and implementation of the simulation-based sustainability toolkit, decision makers may be able to model and evaluate systems of interest for the environmental impacts in a much easier way. The sustainability related performance measures, along with the traditional performance measures, should be able to be returned to the user at the same time to help in decision-making.

3 LITERATURE REVIEW

This chapter summarizes the current body of knowledge relevant to this research. Section 3.1 includes scientific research and results on sustainability issues and metrics. Section 3.2 introduces discrete-event simulation in general. Current simulation software comparison and availability is discussed in section 3.3. The last portion of the chapter reviews some related research work on simulation applied to sustainability.

3.1 Sustainability Related Issues and Measurements

Sustainability, as introduced in section 2, is a pattern of resources usage that “[meets] the needs of the present without compromising the ability of future generations to meet their own needs” defined by the World Commission on Environment and Development (United Nations General Assembly, 1987). It contains three major dimensions. They are economic dimension, social dimension and environmental dimension. This research is primarily focused on the environmental dimension.

From a globe scale, the environmental dimension includes the elements such as atmosphere, ocean, water, land and forest. The main sustainability related issues in atmosphere system are the air pollution. The pollutants of sulphur oxides cause the acid rain. Particulate matters produce the photochemical smog. The chlorofluorocarbons (CFCs) degrade the ozone layer and the increasing generation of greenhouse gases, including carbon dioxide and methane, is very likely to have caused most of the increases in global average temperature since the mid-20th century (IPCC, 2007). Sustainability issues in the ocean system are primarily focused on over-fishing. The lost of forests due to human use is also significant. World Resources Institute reports that 47% of the world’s forests have been lost and the present forests only occupy about a

quarter of the world's ice-free land (WRI, 1998). During the 20th century, the freshwater ecosystems are declining even faster than marine or land ecosystems (Hoekstra, 2006).

In order to quantify the sustainability-oriented performance of human activities, the sustainability metrics have to be identified and established. There exist a vast number of methods and approaches to define the sustainability measures, indicators and metrics. Some research approaches from the ecosystems' perspective and studies the input and output balance of nature. Because markets are not providing the price of many ecosystem services, Costanza et al. (1997) estimate the economic value of 17 ecosystem services, including the gas regulation ecosystem, climate regulation ecosystem, water regulation ecosystem, waste treatment ecosystem, etc., and evaluate the estimated minimum value of the entire biosphere ecosystem to be in the range of US$16 – 54 trillion per year. Zhang et al. (2006) conduct and employ a metabolism model and a harmonious development model, to measure and evaluate the sustainability level in the complex urban ecosystem (CUE) for several major cities in China. Ecological footprint, as a popular sustainability measurement, is a method that estimates the necessary biologically productive area to meet the human demand (Holmberg et al., 1999).

Life cycle assessment (LCA) is a method to measure and evaluate the environmental impact of a given product or service over its entire life span (Curran, 1996). LCA provides an overall framework and numerous studies have been done to develop methods for LCA since 1960s (Matthews et al., 2002). Matthews et al. (2002) analyze the connections between the risk analysis (RA) and LCA and present several methods for incorporating risk analysis into LCA. Hendrickson et al. (2006) introduce another approach, the economic input-output life cycle assessment approach (EIO-LCA), to provide the environmental impact from an aggregate view of the economy sectors that produce all the goods and services in U.S..

Other sustainability measures are studied within a particular industrial system or process. Martins et al. (2007) propose four sustainability metrics, including material intensity, energy intensity, potential chemical risk, and potential environmental impact, to the chemical processes. Jeon and Amekudzi (2005) review and characterize the sustainability measures in transportation systems. In their paper, system effectiveness and efficiency, and the impacts of the system on the natural environment are the two most used measures in transportation systems. Malherbe and Mandin (2007) study the health-risk assessment for volatile organic compounds (VOC) emissions during outdoor ship painting.

3.2 Discrete-Event Simulation Overview

Simulation techniques have been widely used in industry and research areas during the last two decades. The application areas where simulation has been found as a powerful tool include manufacturing systems (e.g. semiconductor manufacturing), inventory systems, computer network application, military applications, logistics and transportation systems, financial decisions and business process and service organizations (e.g. emergency department in hospitals). Recently, simulation techniques have been applied to construction engineering, project management and supply chain management areas. In general, simulation is used to model a complex system numerically where no analytic solution current exists (Law, 2007).

In real-life practice, numerous systems are described as an objective or a discrete set of objectives moving from point to point with one movement at a time, which is referred as the *transaction-flow world view* (Schriber and Brunner, 2008). In those cases, systems change or update only at some particular time within a discrete set of time points. Discrete-event simulation is used to model such a system. In this section, the general mechanism of modeling using discrete-event simulation is introduced.

3.2.1 Discrete-Event Simulation Components

The crucial components in the discrete-event simulation include entities, system states, resources, operations and events. To demonstrate the definition of these components, it is assumed that a system with a single server providing the services to the incoming customer is studied. Using the *transaction-flow world view*, the moving objective in this system is a customer. A customer moves from point to point which includes arriving at the system, requesting the server, capturing the server, releasing the server, and leaving. *Entity* designates a unit of the moving objective, a customer in this case. *System states, or system state variables,* are measurements and indicators of the operation to describe the system at any simulation time. System states can vary for different purposes of study. In the single service example, the system state is defined as the total number of customers in the system. *Resources* describe such modeling components that provide service to the entity. Thus, the server is the resource in the demonstration example. *Operations* are the steps carried out by the entity, including arriving, requesting, capturing, releasing and leaving in this case. Further, *events* are those operations, the occurrence of which changes the system states instantaneously. Since the arriving operation of a customer causes the total number of customers in system to increase by one unit, and the leaving operation causes it to decrease by one, these two operations are events. Thus, in discrete-event simulation, system states are updated by and only by the occurrence of events which are performed by entities, at a discrete set of time points (Law, 2007).

3.2.2 Discrete-Event Simulation Execution

The execution of discrete-event simulation is formed by its nature. Entities take actions by moving to an operation point in the *transaction-flow world view* (e.g. customer arrives at the system) at a discrete set of time points. Thus, if the simulation time reaches one of the time-

points, the simulation model "freezes" the current simulation time and starts to process the entities' movements. This phase where entities accomplish the scheduled movements is introduced by Schriber and Brunner (1998) as *entity movement phase* (*EMP*).

During the entity movement phase, if multiple entities have scheduled movements at the same simulation time point, they all move immediately at no real time. However, the process order for those entities in simulation time depends on the modeling system (e.g. the capturing service movement of the next customer happens the same time as the leaving movement of the current customer, but the current one has to leave first). The current moving entity (real time) is referred as *active state entity* and stored in the *active-entity list*. There is always only one entity in the active-entity list. Other entities, which are scheduled to move at the same simulation time, are stored in the *current events list* waiting to enter the active-entity list. These entities are in *ready state*. Thus, during the entity movement phase, the entity in the active-entity list is processed first, and next entity from the current events list transfers to the active-entity list. The phase continues until all the entities in the current events list are processed.

Entities take actions and change the system states during the entity movement phase. Then, the system stays stable and unchanging until the next event occurs. In order to model the system as simulation time passing by, the general execution is to advance the simulation time to the next earliest event and skip the time in between. This time-advance mechanism is known as *next-event time advance* approach (Law, 2007). And this phase in simulation modeling is introduced by Schriber and Brunner (1998) as *clock update phase* (*CUP*).

To demonstrate the mechanism during the clock update phase, several entity states and entity lists are first introduced. Entities are delayed in a simulation model, either for a given amount of simulation time or for satisfying some specified conditions. In the former case, entities

are in the *time-delayed state.* In the later case, entities are in the *condition-delayed state*. The main difference between these two delay states is that the delay time is known for entities in time-delayed state and is undetermined for entities in condition-delayed state. Entities in time-delayed state are stored in the *future events list.* The list is unique. Entities stored in the future events list have their own known moving time in the future. Entities are ordered from top to bottom in such a way that their moving time is increasing. Entities in condition-delayed state are stored in the *delay list.* A simulation model contains one or multiple potential delay lists. Each one of the delay lists corresponds to one particular condition. During the clock update phase, simulation system checks the future events list and delay list. If the entities in the corresponding delay list are scheduled to move, either their moving time is reached or their specified conditions are resolved, they will be transferred into current events list.

After completing the CUP, simulation system enters EMP to process the current event list. The loop between EMP and CUP continues until the ending simulation condition appears. To summarize how discrete-event simulation models execute, the steps in the two main phases are listed as follows. The entity movement phase includes the tasks:

- Accomplish all current events in the current events list;
- Update the system states, system performance measures;
- Update the future events list and delay lists;
- Go to clock update phase if the simulation does not end.

The clock update phase consists of:

- Move the simulation clock to the next earliest event time;
- Update the current events list by removing the entities from the future events list and delay lists to the current events list.

- Go to entity movement phase if the simulation does not end.

Figure 3-1 shows the flow chart of the execution in a single simulation run (Schriber and Brunner, 1998).

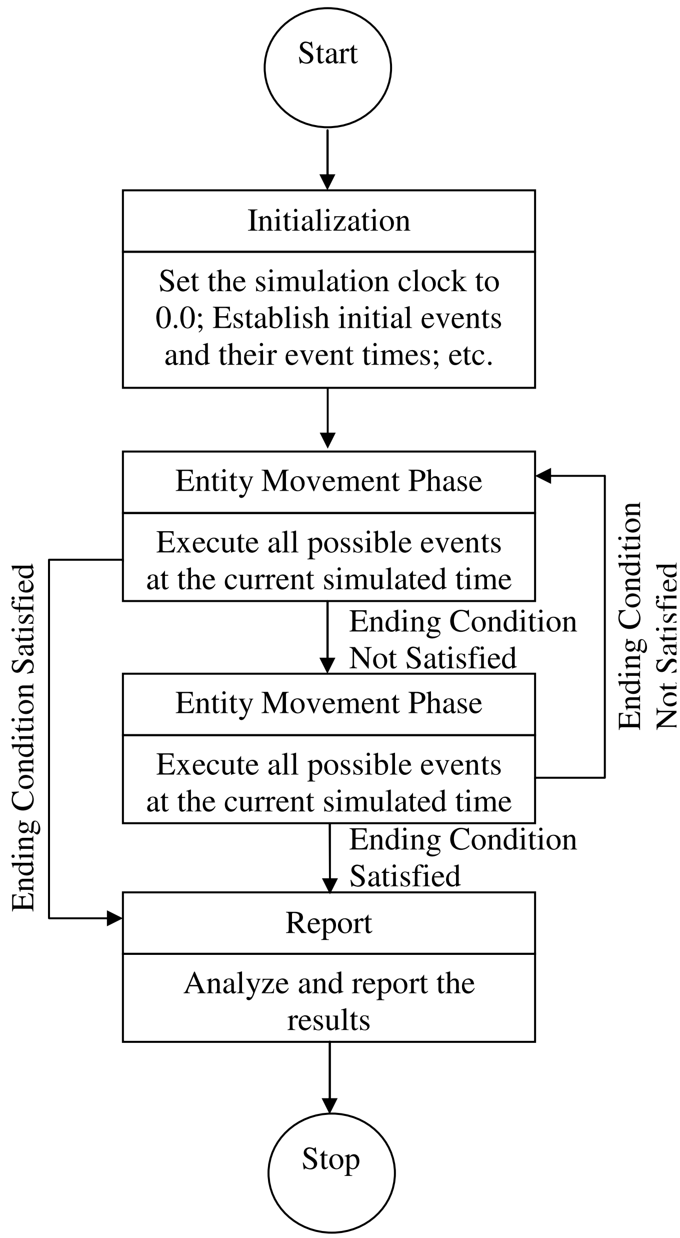


Figure 3 - 1: Simulation Execution of a Single Replication (Schriber and Brunner, 1998)

3.2 Simulation Software

These exists a vast amount of software that is available for discrete-event simulation, which serves for the commercial purpose and the research purpose as well. According to the biennial survey of discrete-event simulation software in 2007 conducted by INFORMS, there are more than forty tools for discrete-event simulation (Swain, 2007).

The current simulation software is categorized into two major groups. They are general-purpose simulation software, and application-oriented simulation software. The general-purpose simulation software is able to solve various discrete-event simulation systems. The application-oriented simulation software is used for one particular type of system, including manufacturing system, supply chain system, process reengineering system, contact center system, health care system and animation (Law, 2007 and Banks, 1998). Some of the commonly used general-purpose simulation software includes ARENA (Rockwell, 2005 and Kelton et al., 2006), Extend (Imagine That, 2006), and GPSS/H (Henriksen and Crain, 1998).

In the application-oriented simulation software category, AutoMod, ProModel and WITNESS are used for manufacturing system. ProcessModel and SIMPROCESS are designed to model process reengineering system. MedModel is a simulation tool that is used in health care area. Proof Animation serves as a general-purpose animation package, although most simulation animators are integrated into its software.

3.3 Simulation Applied to Sustainability

Given the increased focuses on sustainability, simulation modeling has been used in the recent research to analyze systems of interest while taking into account of the environmental impacts. From a systems perspective, Kurkovsky (2004) analyzes the indictors of sustainability, as well as three major simulation models that are used to evaluate the sustainable development in

three different regions, including Japan, Germany and USA. In addition to the three major simulation models, Kurkovsky also proposes a problem-based simulation methodology for regional sustainable development assessment.

Simulation techniques have been applied to particular systems for solving the sustainability issues. Blanco et al. (2005) have applied simulation to forest management by introducing a process-based simulation model to evaluate temporal changes in ecosystem nutrient dynamics of managed and non-managed forest stands. Ruwanpura et al. (2003) propose a simulation model to determine solid waste that is generated at a construction site. The paper also provides a protocol of using the simulation results to help with the planning of construction projects. Halima and Srinivasan (2008) develop a simulation-optimization framework to identify and evaluate sustainable alternatives for batch processes. The simulation-optimization approach contains a process simulation model and a multi-objective stochastic optimization method. The waste generation due to specific system variables levels is simulated and integrated with the optimization method. Yu et al. (2009) develop an agent based simulation model to help in decision-making for Hawaii's longline fishery activities. In addition, simulation models are used in the context of logistics and transportation related to sustainability. Cortes et al. (2008) utilize simulation to estimate energy consumption and emissions for transportation networks in urban and rural areas over time horizons on the order of 10 years. Barth and Todd (1999) analyze the shared vehicle systems and their resulting emissions using a simulation model. Likewise, Lee (2000) constructs a dynamic network model to study the impact of traffic flow control on carbon dioxide emissions of passenger cars. In each of these cases, specialized models are designed for a specific purpose requiring significant time and effort. Simulation models are also used in manufacturing to evaluate sustainability issues. Heilala et al. (2008) introduce sustainability

measurements for designing manufacturing systems. Energy efficiency and CO_2 emissions are the two main environmental impacts they consider in the paper. They integrate the evaluation of these two measures into factory simulation software.

Wohlgemuth et al. (2006a) develop the software *Milan* which combines the discrete-event simulation and material flow analysis together. They propose a component-based approach for integrating the two methods and develop the plug-ins under *Milan* framework. Some of the plug-ins are developed for simulation purposes including random variates generation, event list management, etc., which are used for modeling the industrial process and material usage. The results obtained from the simulation-based plug-ins are analyzed by the components that are built for material flow analysis, which intends to evaluate the sustainability aspects of energy and material flows in production processes (e.g. material consumption, emission generation). The advantage of the approach is integrating modeling an industrial system and analyzing its environmental impacts together in one single model. However, the work is mainly focusing on the coding of the plug-ins for discrete event simulation and the administration of an existing material flow analysis tool *Umberto*, while little attention is given to research and evaluate the sustainable aspects and environmental impact measures for the systems under study. Wohlgemuth et al. (2006b) apply the material-flow simulator *Milan* to evaluate the economic and ecological aspects of inventory management strategies. The customized simulator components permit reusability and flexibility for modeling complex inventory system configuration. However, in the paper, the environmental indicators used to evaluate the emissions impact (e.g. CO_2 generation) are not directly set up to be the actual amount of emissions generated. The paper chooses indirect indicators, such as the amount of deliveries and the reorder intervals, to describe the emissions generation impact.

The current literature in simulation-based sustainability tools development is limited. There is a need to develop a flexible framework for efficiently modeling and analyzing systems that require environmental performance evaluation, in addition to performance measures that are traditionally oriented such as work in process, time in system, utilization, and cost.

4 RESEARCH SCOPE AND METHODOLOGY

When using simulation for modeling and analyzing system performance, traditional performance measures, such as productivity and efficiency, are frequently used. Some of the commonly used performance measures, such as processing time, time in system, and resource utilization, are built-in for most of the simulation platforms. This research intends to supplement the traditional performance measures with the environmental focused measures and make the collection of the environmental measures as easily as other traditional performance measures. A simulation-based sustainability toolkit will be developed for this purpose. In this section, we introduce the scope of the research, as well as the general research strategy.

For the three aspects of sustainability, social, economic and environmental aspect, the long term research plan is to design a sustainability toolkit (or a set of toolkits) that takes an integrative systems approach to modeling problems from all three fields. This thesis paper focuses primarily on the environmental aspect of sustainability. Compared to the other two aspects of sustainability, the environmental aspect studies more immediate impacts of the sustainability issues. For instance, the amount of air pollution emitted into the air is the first direct outcome of an industrial process. The environmental aspects of systems have been studied and literature exists for many applications for determining and measuring environmental impacts. In the future, it is hoped that the sustainability toolkit can be extended to also aim in decision making related to the social and economic aspects of sustainability, and include the measurements for human health and economic impact.

In order to quantitatively measure the environmental impacts, studies have been carried out to obtain the metrics of the environmental impacts. Some of the metrics to quantitatively measure environmental impacts include ecosystem oriented metrics, product and service oriented

metrics and process oriented metrics. Of these, discrete-event simulation modeling is well suited for studying the process oriented metrics within a dynamic system. Therefore, the process oriented metrics for sustainability will be the focus of the development of the simulation toolkit.

In order to design and develop such a sustainability toolkit, a general research strategy is developed. The following research steps explain the research strategy.

- *Develop a sustainability toolkit development methodology*: The methodology answers the question: given a selected system/process, how to develop a sustainability toolkit for the system/process of interest. Simulation state variables and simulation events are indentified. Mathematical relationships are defined. The development methodology is used throughout this research.
- *Identify application areas of the sustainability toolkit*: Since the scope of this research is to apply process oriented sustainability metrics to develop simulation based sustainability toolkit, the potential systems/processes are identified where the toolkit can be useful. This research develops the toolkit based on the systems/processes selected.
- *Implement the development of the sustainability toolkit*: After applying the development methodology to a selected system/process, the sustainability toolkit needs to be implemented and constructed in a simulation software platform.
- *Perform verification, validation and testing of the sustainability toolkit*: After the toolkit is implemented in a simulation platform, the verification, validation and testing of the toolkit will be carried out in order to evaluate the capabilities and limitations of the toolkit.

The following subsections explain in detail these four research strategy components.

4.1 Sustainability Toolkit Development Methodology

To develop such a sustainability toolkit in general and provide the environmental measures, a toolkit development methodology is designed. The methodology consists of the following steps:

1. Define the system to be studied;
2. Define sustainability performance measures;
3. Define simulation state variable;
4. Determine simulation events;
5. Formulate mathematical, statistical, and logical relationships among the state variables; and
6. Toolkit implementation.

In particular, the details of the toolkit development methodology include the following. In step 1, in order to construct and use such a simulation-based toolkit, a system or process is first selected to study. The system or process under study has environment-related issues that need to be addressed and analyzed to help in decision-making.

The second step identifies the sustainability factors for the selected system. System factors that cause the environmental impacts are identified. For instance, an ocean fishing system may have the potential over-fishing as a sustainability factor with an environmental impact, while a manufacturing system may have concerns about the energy consumption and waste and emissions generation. Step 2 answers the 'what' and 'why', which are 'what environmental statistics are people interested in collecting and why are they important'. The following steps answer in details for the 'how', which is 'how to collect the environmental statistics'.

Step 3 of the methodology involves determining simulation state variables to describe the sustainability factors and environmental impact measures in the corresponding system. In discrete-event simulation, system state variables are the indictors of system performances. For instance, traditional system states variables may include work in process, flow time, and waiting number in queue, etc. In terms of sustainability factors, system states variables, which represent quantitatively for the environmental performance, differ from other traditional state variables and are defined separately in this step.

Step 4 involves identifying the sustainability factors and environmental impact performance measures for the selected system. Events in the system are the occurrences that change the corresponding traditional or sustainable system state variables. Thus, the methodology analyzes the system to indentify each event that initializes or updates the environmental impacts. Since discrete-simulation is used, the changes to the state variables occur upon the instantaneous occurrence of simulation events. For the sustainability measures such as emissions which occur continuously over time, it is assumed, modeling in discrete-event simulation, that emission quantity between the occurrences of events can be represented as a function of time, a probability distribution or a stochastic process.

Once the system state variables and the events that cause the changes to the state variables are identified, step 5 in the methodology is answering how the environmental impacts are measured quantitatively. The relationship is formulated mathematically and logically. There exist a number of published scientific studies which contains this information, including papers and some open resources such as U.S. Environmental Protection Agency (EPA). This step also includes translating the relationship, whether it is mathematical formulations or logical interactions, into the simulation model and allowing the model to update the state variables.

In the last step, the system state variables, the simulation events, and the mathematical and logical relationships are integrated into a sustainability toolkit which can be used directly for evaluating the sustainability performance. The implementation of the toolkit is discussed in section 4.3.

A general methodology for developing a sustainability toolkit is introduced in this section. The methodology describes certain steps to take to construct the toolkit for a system of interest. The methodology is used throughout this research to develop the sustainability toolkit.

4.2 Application Areas of the Sustainability Toolkit

Environmental sustainability issues appear in many areas. This research focuses on common industrial systems or system components where there is a need to analyze the environmental issues. In the selected systems, further study is performed in order to identify what are the environmental factors and how to represent them. Table 4-1 displays a list of potential systems and system components, as well as the environmental impacts for each of the systems. This research designs the sustainability toolkit to analyze environmental performance measures for the systems in Table 4-1. Details of the toolkit representing each of the systems are fully discussed in section 5.

Logistics and transportation systems are systems dealing with distribution. Given the customers' need, logistics and transportation systems are set up in order to try to meet a certain customer service level in the most efficient way. Simulation has been used to model the distribution models such as dock operations (Banks, 1998). Since logistics and transportation systems contain transportation sources such as vehicles, trucks, ships and airplanes, the associated environmental impacts are mainly arising in fuel consumption and air pollution generation. The increased generation of carbon dioxide (CO_2), also known as greenhouse gas

(GHG), is becoming a worldwide environmental issue. Based on the report from EPA, transportation section is the fastest-growing section of GHGs in U.S., accounting for 47 percent of the net increase in total U.S. emissions since 1990 (EPA, 2006). Other air pollutions may also include carbon emissions (such as CO), nitrogen oxides emissions (such as NO_2) and total hydrocarbon (THC). As a result, evaluating the emissions generated in a logistics and transportation system, especially from running the transportation vehicles, is useful to help in decision-making such as determining the route assignments. A toolkit that deals with the collection of the emissions specifically is preferred when constructing the simulation model. Section 5.1 introduces the sustainability toolkit modules that are constructed for logistics and transportation systems.

Table 4 - 1: Systems and System Components applied to the Sustainability Toolkit

Systems and System Components	**Environmental Issues and Impacts**
Transportation and Logistics System	• Fuel Consumption • Air Pollution Generation
Industrial and Manufacturing Processes	
→ Industrial Coating Process	• Paint and Solvent Consumption • Volatile Organic Compound (VOC) Generation
→ Injection Molding Process	• Energy Consumption • Emissions and Wastes Generation
→ Plastics Processing	• Energy Consumption • Water Usage • Emissions and Wastes Generation
Material Handling System	• Air Pollution Generation

Using simulation as a tool to evaluate industrial and manufacturing systems has a long history and simulation is ranked number two for the most widely used techniques in industry (Law, 2007). However, traditional system performance measures in manufacturing systems include productivity, efficiency and cost but not much sustainability-related attention is given. Systems are mainly studied in order to increase the throughput, decrease the lead time and increase the utilizations of recourses (e.g. machines, workers). Traditional system performance analysis using simulation includes throughput analysis, time-in-system analysis and bottleneck analysis. Sustainability related issues, which mainly contain the raw material consumption, energy consumption and waste and emission generation, are becoming a part for decision support and thus require the sustainability toolkit to evaluate the environmental impacts within the manufacturing system. For instance, semiconductor manufacturing is a very complex process and involves in many environmental related issues such as the generation of air toxins or hazardous air pollutants (HAPs), water usage, waste water generation, chemical material usage, and chemical waste generation. Table 4-1 lists some industrial processes under study including industrial coating process, injection molding process and plastics processing. The paint spray process in a manufacturing system generates the indoor air pollution such as volatile organic compounds (VOCs). VOCs are organic chemical compounds which have significant vapor pressure (0.01 kPa or more). The emitted VOCs lead directly to damage the environment and human health (Rentz et al., 2002). The injection molding process and plastics processing also generate air pollutions such as CO_2 and SO_2, known to cause acid rain. Energy consumption is another sustainable issue appearing in both systems. Section 5.2 introduces a toolkit module which analyzes the listed three industrial and manufacturing systems.

The distribution warehouse plays a crucial role of developing efficient and low-cost operations. Material handling equipment, such as automatic guided vehicle (AGV) and forklift, is often used for transporting materials such as raw materials, work in process products and finished products. Vehicles generate air emissions, and require battery charging while they are working. Hence, the potential environmental impacts include evaluating the emissions generated and the energy consumption for using batteries. In 2007, EPA estimates that forklift operations generate 297,973 tons of nitrogen oxides (NOx), 64,892 tons of hydrocarbon (HC) and 1,357,677 tons of carbon monoxide (CO), which shows that forklift operations is a large contributor to the air pollutions. Section 5.3 introduces a set of toolkit modules that can model and analyze emissions generated from a warehouse material handling system.

In Table 4-1, sustainability toolkit and modules can be developed for each of the systems separately. A system-independent toolkit generalization approach features a general sustainability toolkit that can be utilized to various systems to collect environmental performance measures. Commonly used logic and statistical flow among the system-dependent sustainability toolkits will be identified and studied. Section 5.4 introduces a general toolkit which can be applied to all kinds of systems to collect user-defined sustainability performance measures.

4.3 Development and Implementation of the Sustainability Toolkit

After the system is selected, toolkit development methodology is applied to the studied system. The environmental performance measures and the state variables are defined, the simulation events are identified and the mathematical relationships are established. The toolkit is then implemented in simulation software platform. Various simulation packages have different levels of flexibility. Appropriately chosen simulation software can reduce the simulation time, provide an easy implementation of the toolkit development, and enable ease use of the toolkit.

In this research, ARENA simulation software by Rockwell Automation Technologies is used for the implementation of developing the sustainability toolkit. The ARENA simulation software has been selected because of the following characteristics:

- *General-purpose simulation software*: This research intends to evaluate and develop the sustainability toolkit for various systems of interest including but not limited to the systems listed in Table 4-1. Therefore, compared to application-oriented simulation, general-purpose simulation software is chosen to implement the sustainability toolkit development because it can be used to model various systems of interest. ARENA is one of the widely used general-purpose simulation packages.
- *Ability for the user to create user-specified template*: ARENA has its own template development tools built into the software itself, which allows the users to create their own templates and modules. It provides the flexibility of designing a separate sustainability template. The modules in the sustainability template, which perform the collections of sustainability performance measures, can be used along with other default traditional modules in a drag-and-drop manner to model a system.

4.4 Verification, Validation and Testing of the Sustainability Toolkit

In order to make the sustainability toolkit creditable and trustable to the user, the toolkit needs to be verified, validated and tested for usage. The following steps are carried out first, and then the verification, validation and testing of the toolkit are performed.

1) **Set up System Configuration.** In the selected system of study, define a system configuration in order to be modeled. The system configuration should be set up and specified in such a way that the desired sustainability performance measures will be collected.

2) **Construct Original Simulation Model.** Construct a simulation model using the system configuration specified. Use necessary information including the logical relationship, the mathematical relationship and the statistical relationship to collect the desired environmental performance measures.

3) **Reconstruct Simulation Model using the Sustainability Toolkit.** After the toolkit is built using the development methodology, the system is remodeled using the toolkit. The system configuration is kept the same as for the original simulation model.

4.4.1 Verification of the Toolkit

Verification of the sustainability toolkit contains two parts. One part is performing the verification of the simulation model using the toolkit. It determines whether the conceptual system model has been translated into a correctly working computer program. The other part of the verification of the toolkit is determining whether the methodology of developing such a toolkit has been successfully translated into a computer program.

The verification of the simulation models have several approaches which are to:

- Debug the simulation models, one of which does not use the sustainability toolkit and the one does use the toolkit;
- Present a structured walk-through of the simulation programs;
- Observe the animation of the simulation to determine whether it is reasonable; and
- Check model results for reasonableness.

The verification of the sustainability toolkit itself cannot be separated from the verification of the simulation model using the toolkit. However, the toolkit verification should contain the following additional steps which are to:

- Debug the programming of the toolkit;

- Check the toolkit compatibility with the models using the toolkit; and
- Examine the toolkit for returning the desired sustainability performance measures.

4.4.2 Validation of the Toolkit

In general, validation of a simulation model examines whether the model behaves close enough to the real system. It is the process of determining whether a simulation model is an accurate representation of the system under study (Law, 2007).

Since the simulation model is only an approximation of the system, the validity increases as more details are provided by the system. This research focuses on the development of the sustainability toolkit, the mathematical relationships used in developing the toolkit is obtained from other research and study. Therefore, the validation of the toolkit is conducted against the referenced scientific papers. The validation of the toolkit should include the following aspects:

- Use the literature which contains as much information as possible of the environmental impact measures and relationships;
- Document the assumptions made when obtaining the relationship and building the toolkit;
- Review and evaluate the assumptions; and
- Compare the results from the simulation model with the corresponding results obtained from the chosen scientific studies, or historical data.

4.4.3 Testing of the Toolkit

Once the sustainability toolkit has been developed and implemented, the testing of the toolkit needs to be performed in order to make sure that using the toolkit will not change the results. The toolkit provides the user with the ability to easily collect the environmental

performance measures as readily as traditional performance measures. The toolkit will not affect the simulation modeling from the operational point of view.

Therefore, the testing of the toolkit mainly is to compare the results from running the simulation model without the toolkit and the one using the toolkit. No statistical differences should be found on the mean value of the obtained performance measures when running the two simulation models in their steady states.

5 DEVELOPMENT OF SYSTEM-BASED TOOLKIT MODULES

In this section, the development and the implementation of the sustainability toolkit is applied to systems including logistics and transportation system, industrial processes system, and warehouse material handling system. Section 5.1, 5.2 and 5.3 describe how the sustainability toolkit is constructed for each of the three systems. Section 5.4 describes the design of a general toolkit which is applicable to the systems under study. The main objectives of the toolkit are:

- *Toolkit collects desired sustainability-related statistics.* This is the main functionality that the toolkit provides.
- *Toolkit has a user-friendly interface where the user can input parameters easily.* The toolkit is displayed to the user through one or multiple interfaces. The interface is constructed using elements such as text boxes, drop-down menus and commands which are very user-friendly. The interface increases the easy for using the toolkit.
- *Toolkit is integrated into platform simulation package and can be easily used in the simulation package.* The toolkit is implemented in the target simulation software and it is used in the same manner along with other modules of the platform simulation package.
- *Toolkit reports the collected sustainability-related statistics as readily as traditional statistics.* By default, the traditional performance measures such as productivity and efficiency are reported automatically at the end of the simulation. The sustainability toolkit developed is able to also report the collected sustainability-related statistics automatically at the end of the simulation, along with other traditional statistics.
- *The general toolkit modules can model various systems with user-defined sustainability measures.* As mentioned in section 4.3, the sustainability toolkit is generalized into one

unique toolkit with the ability to analyze the environmental measures in all kinds of systems.

Figure 5-1 depicts the ARENA template that is created representing the sustainability toolkit. The modules created in this template are applied to the listed systems and processes. The modules are also used in conjunction with the other default templates and modules to conduct simulation modeling.

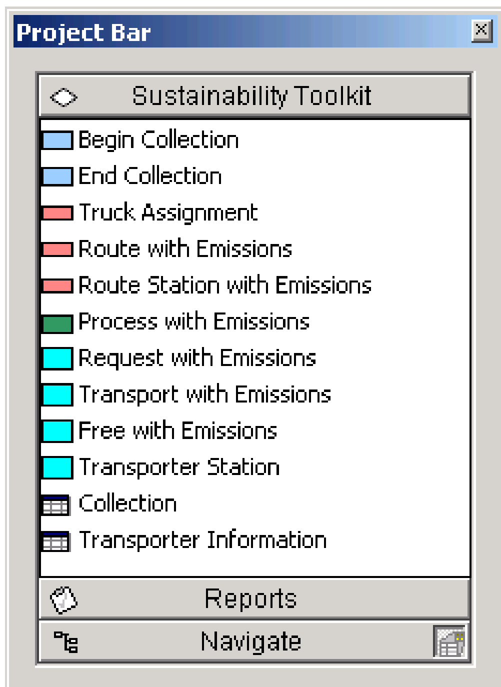


Figure 5 - 1: Overview of the Developed Sustainability Toolkit

5.1 Transportation and Logistics Modules

Transportation and logistics systems are selected to be modeled because of their frequent application in simulation modeling. Transportation and logistics system often result in massive air emissions and energy usage, which makes it a potential candidate for utilizing the sustainability toolkit to aid in decision-making considering the environmental impacts within the system. In Kuhl and Zhou (2009), the concept of modules for modeling transportation and

logistics systems is introduced. Since then, more flexible versions of the modules have been developed. The details of the development are provided here for completeness.

To demonstrate the toolkit development methodology, the methodology is applied to typical logistics and transportation systems. In particular, the environmental factors and performance measures for the transportation system are identified, system state variables are determined to represent the factors and the measurements, system events that change the system state variables are specified and the mathematical relationship for the environmental factors and the system state variables is established and updated at the occurrence of each event. An implementation of the toolkit development for a particular transportation problem is then conducted. To illustrate the application of the toolkit, a simulation model is constructed for the following transportation system example.

Example 5-1: *Suppose a company has a centralized warehouse facility from which it makes deliveries to customers on a daily basis. The company currently has delivery trucks that consume diesel fuel. Each day routes are assigned to trucks. The route may consist of a variable number of delivery stops at customers and variable route distances that can each be represented by a probability distribution. The company is currently in a position where it needs replacing part of its fleet of trucks. The company wants to determine the number and type of trucks to purchase (e.g. capacity, type of fuel, and engine power) while taking into account the energy consumption and emissions in addition to traditional system performance measures such as cost and customer service level. A sample instance of truck routings is displayed in Figure 5-2.*

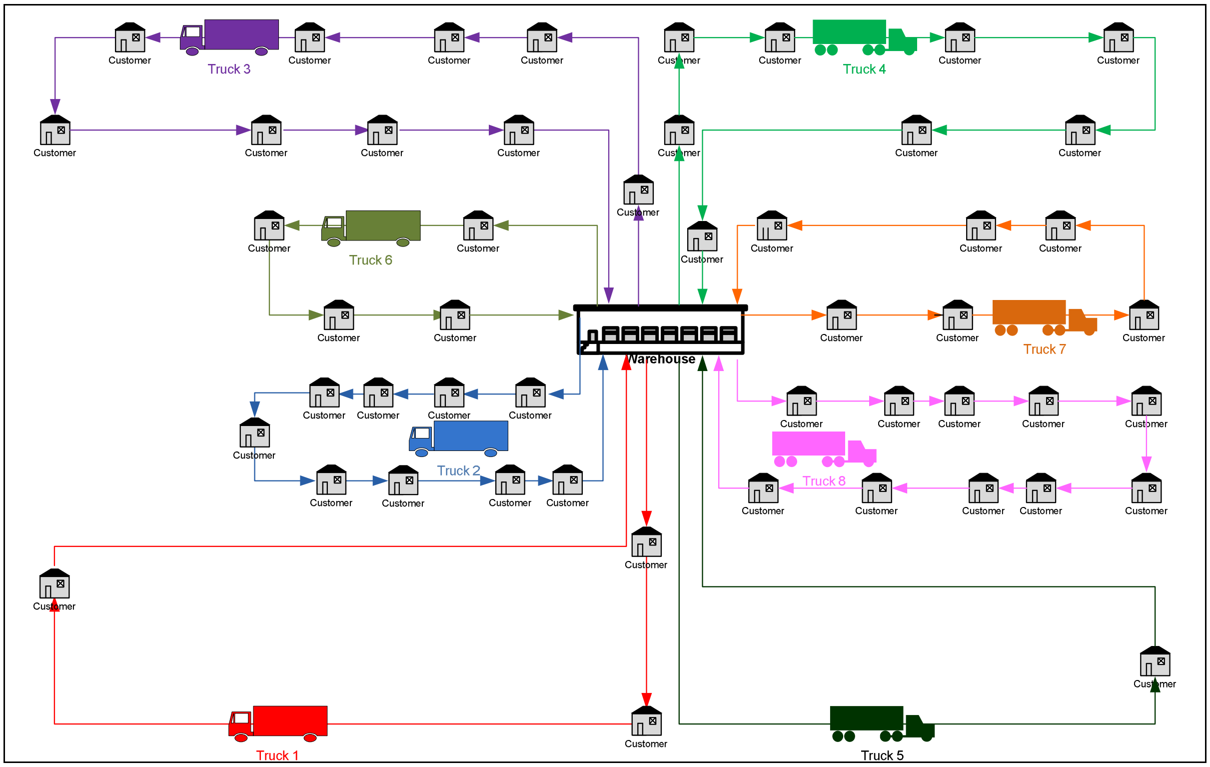


Figure 5 - 2: Diagram of a Sample Instance of Truck Routings

This example depicts a system where simulation could be an appropriate tool to evaluate various system configurations in order to make a good decision. The prototype toolkit developed in this section allows the user to evaluate the system using traditional system performance measures such as routing time, routing distance and customers served, as well as sustainability performance measures.

The general simulation-based sustainability toolkit development methodology is applied to the stated logistics and transportation system. Each of the steps in the methodology is considered and analyzed for the logistics and transportation system. The following sections describe the detailed development steps.

Select the system. Light trucks contain pick-ups, minivans, SUVs, and vans. Medium and heavy trucks are commonly used trucks for logistics and transportation companies (Davis et al., 2009). Most medium and heavy trucks use four stroke turbo intercooler diesel engines and

consume diesel fuel in United States. According to Davis's report, in 2006 a total amount of 4,649 trillion Btu energy is consumed by medium and heavy trucks using diesel, which is more than 75% of the overall energy consumed in both highway and nonhighway transportation system. The burning of diesel fuel generates gaseous residues. As reported in U.S. Carbon Dioxide Emissions (EIA, 2009), diesel fuel accounts for 23.2% of the transportation sector's CO_2 emissions in 2008, while transportation sector CO_2 emissions have risen by 21.1% since 1990.

Identify the environmental performance measures. As the energy consumption and emissions generation issues becoming increasingly important, the main sustainability aspects in the logistics and transportation system include the fuel consumption, as well as the air pollution emitted by operating the trucks. These two factors are both quantitative factors. Therefore, the performance measure for the fuel consumption is measured by the usage of the fuel for each of the delivery trucks. The air pollution emitted by operating a truck, depending on the different types of emissions that the user is interested in analyzing (e.g. carbon monoxide, carbon dioxide), is also collected by each of the trucks during one particular trip. The sustainability concern of modeling the transportation problem is to evaluate the amount of fuel consumed and the amount of emissions generated.

Define simulation state variables. Two assumptions of the transportation system are made including the following:

1. Truck is travelling at the average speed; and
2. Emissions generated per unit of time are the same regardless of the truck status (travelling or idling).

In order to represent the system performance measures including the traditional performance measures and the environmental performance measures, corresponding state

variables are specified and defined separately. Table 5-1 lists the system's parameters used in the logistics and transportation system.

Traditional system performance measures usually include the system productivity and efficiency. In a typical transportation problem, the delivery transporter (e.g. truck) moves to its destination (e.g. customer), makes a stop, and moves to the next destination. The productivity measures are then represented by the total number of stops (S_{Truck}) made by each truck in a route. The efficiency measures are focused primarily on the transportation phase in the system. Since truck represents the delivering equipment, the efficiency measures consist of the travel distance and the travel time of the trucks. Accordingly, the state variable that represents travel distance is the total route distance of each truck (D_{Truck}) and the state variable that represents travel time is the total route time of each truck (T_{Truck}). Additionally, corresponding state variables are set up as the total stops made in the system (TS_{System}), total route distance in the system (TD_{System}) and total route time in the system (TT_{System}).

The environmental impacts, including fuel consumption and emissions generation, are specified as the sustainability performance measures. Thus, two sets of simulation state variables need to be established. Fuel type (Z) is used by the delivering truck, the environmental impact of fuel consumption is measured in the total amount of fuel used by each truck ($F_{Z,Truck}$). If the user is interested in knowing the emission type (Y) that has been generated by the delivering truck, the environmental impact of emissions generation is measured in the total amount of emission emitted by each truck ($E_{Y,Truck}$). Parallel to the traditional performance measures, system fuel consumed of the chosen fuel type ($TF_{Z,System}$) and system emission generated of a particular emission type ($TE_{Y,System}$), are used as the system level measurements among all delivering trucks. Table 5-2 summarizes the system state variables defined.

Table 5 - 1: Parameters for Logistics and Transportation System

Parameter Notation	Parameter Meaning	Parameter Unit
t	Engine Running Time	seconds, minutes, hours, days
d	Truck Traveling Distance	miles
s	Truck Average Speed	miles/hour
h	Truck Engine Horsepower	horsepower
e	Truck Efficiency	miles/gallon
Z	Fuel Type	----
Y	Emission Component	----
CE_Y	Collection of Y	0 (no collection) or 1 (collection)
$C_{Y,Z}$	Emission Coefficient	grams/horsepower · hour
Idle	Truck Idling	0 (truck off) or 1 (truck idling)
Truck	A specific Truck	----
System	Entire Transportation System	----

Table 5 - 2: State Variables for Logistics and Transportation System

State Variable Notation	State Variable Definition	State Variable Unit
S_{Truck}	Total Number of Stops made by Truck	count
D_{Truck}	Total Route Distance traveled by Truck	miles
T_{Truck}	Total Route Time by Truck	hours
TS_{System}	Total Number of Stops made by System	count
TD_{System}	Total Route Distance traveled by System	miles
TT_{System}	Total Route Time by System	hours
$F_{Z,Truck}$	Total Fuel Z Consumed by Truck	gallons
$E_{Y,Truck}$	Total Emission Y Generated by Truck	grams
$TF_{Z,System}$	Total Fuel Z Consumed by System	gallons
$TE_{Y,System}$	Total Emission Y Generated by System	grams

Determine the simulation events. To model the system, the primary source of energy consumption and emissions, in the stated logistics and transportation problem, are caused by the trucks. The general performance measures such as routing information and stops made are reflected by trucks as well. Thus, a truck represents an entity in the simulation model, which flows though the system and controls the delivery process.

The simulation model consists of three types of events, the creation of the truck entity, the arrival of a truck at a stop and the departure of a truck from a stop. The creation of trucks introduces the truck entity into the simulation system. Although the truck is consuming the fuel

and generating the emissions continuously in time, the problem is relaxed and modeled as a discrete-event simulation. The simulation model only collects and updates the consumption of the fuel and the generation of the emissions at the time point when an event occurs. For instance, when arriving at a destination, the truck collects the emissions which are generated during the route from its previous stop. Discrete-event simulation models this process as if the emissions are generated at one single time point. Then the truck can be left either idling or turned off when it remains at the stop. The truck continues to generate the emissions if left idling. However, the truck stops generating emissions when turned off. In both cases, the system waits until the occurrence of the next event to change the system state and record corresponding emissions. Similarly, the occurrence of the departure event causes the system state variable to update as well. The number of stops is increased by one unit. The emission variable is increased by the amount of emission that is generated during the time period when the truck remains at the stop.

In the logistics and transportation system, the traditional performance measures, which are number of stops made, truck travel distance and truck travel time, are simply accumulated and recorded when the arrival event or the departure event occurs. For instance, when the departure event occurs, the variable of total stops made by each truck is incremented by one unit for that particular truck. When the truck arrives at a new destination, the variable of the truck routing distance is accumulated by the distance between the current stop and the previous stop. Similarly, the variable of the truck routing time is accumulated by the time that the truck spends from the previous stop.

Formulate the mathematical relationship. The primary environmental performance measures in the logistics and transportation system are the fuel consumed and the emissions generated by trucks. As fuel is burned, emissions are produced. Manicom et al. (1993) conduct a

study on the emissions produced by trucks of varying horsepower consuming diesel and four variations of soya blend biodiesel. The emissions which are studied in the paper consist of gaseous residues including carbon monoxide (CO), carbon dioxide (CO_2), nitrogen oxides (NO_x) and total hydrocarbon (THC). The quantity of emissions (*f*) generated from burning the fuel, which is analyzed in the paper as well, is a function of several parameters including the type of the fuel (*Y*), the truck engine horsepower (*h*), and the time that the engine is running (*t*). The results of this research study provide a list of emission coefficients ($C_{Y,Z}$), in grams per horsepower per hour, for various fuel types including diesel, and 10%, 20%, 30%, and 40% soya blend biodiesel. The relationship is specified as

$$f_Y(t) = C_{Y,Z}ht \qquad \text{(Equation 5 - 1)}$$

where the emissions produced *f* of emission component *Y* over time interval *t* is equal to the emission coefficient $C_{Y,Z}$ for emission component *Y* given fuel type *Z* times the truck horsepower *h* times the time period length *t*. The function can be used directly when the truck is idling. When the truck is traveling, the time component can be calculated as the distance traveled *d* divided by the average speed *s* resulting in the relationship function

$$f_Y(d) = \frac{C_{Y,Z}hd}{s}. \qquad \text{(Equation 5 - 2)}$$

If a truck has multiple stops in a single route, the total emission type *Y* generated in the route is calculated as the summation of the emissions generation within all time periods, in which t_i represents the time interval between stop (*i*-1) and stop *i* while τ_j represents the time that the truck spends at the stop *j*. Thus,

$$E_{Y,Truck} = \sum_{i=1}^{S_{Truck}} f_Y(t_i) + \sum_{j=1}^{S_{Truck}} f_Y(\tau_j).$$

With the knowledge of the truck efficiency e, which is specified as the average distance traveled using one gallon of fuel type Z, the amount of fuel consumed g is calculated as the distance traveled d divided by the truck efficiency e, resulting the following relationship

$$g_Z(d) = \frac{d}{e}.$$

Similarly, if d_i represents the distance between stop (i-1) and stop i, the total amount of fuel consumed in one single route for each truck equals to the summation of the fuel used among all the stops, resulting in

$$F_{Z,Truck} = \sum_{i=1}^{S_{Truck}} g_Z(d_i).$$

Parts of the performance measures, including both the traditional and the sustainability performance, are collected by each individual truck (e.g. total CO generated for one truck), while some of them are collected in the system level (e.g. total CO generated within the system). To obtain the system level performance measures, the corresponding measures for each truck are added together, such as

$$TF_{Z,System} = \sum_{Truck} F_{Z,Truck},$$

$$TE_{Y,System} = \sum_{Truck} E_{Y,Truck}.$$

The implantation of the state variables, relationships and events into the transportation and logistics modules are introduced in the next part.

5.1.1 Transportation and Logistics Modules Development

After identifying the simulation state variables, entities, events and the mathematical relationship, the sustainability modules are constructed and implemented for the transportation

and logistics system. Figure 5-3 shows an overview of the transportation and logistics modules developed.

The sustainability toolkit for the transportation and logistics consists of three modules including *Tuck Assignment* module, *Route with Emissions* module, and *Route Station with Emissions* module. Figure 5-4 displays the user interfaces of each of the three modules. *Truck Assignment* module is where the user specifies the truck parameters such as truck horsepower, truck efficiency, average speed, fuel capacity and the fuel typed used. The user also chooses the emission components to be collected at the end of the simulation. *Route Station with Emissions* module represents a physical location. The probability of the truck idling while it remains at the station is specified in this module. *Route with Emission* is the module used to route the truck to its destination location, represented by the *Route with Emissions* module.

To demonstrate the sustainability toolkit development process, a specified logistics and transportation problem has been defined. In the studied system, the company has eight delivery trucks. The trucks are sent out for work on a daily basis. Each of the trucks serves a set of customers every day. The truck is routed to the customer, delivers the products with the truck idling or turned off, moves to the next customer and eventually returns to the warehouse.

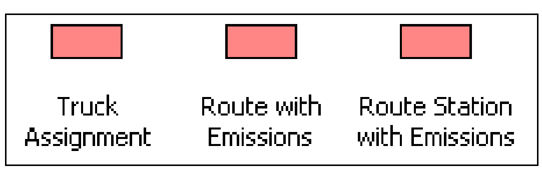
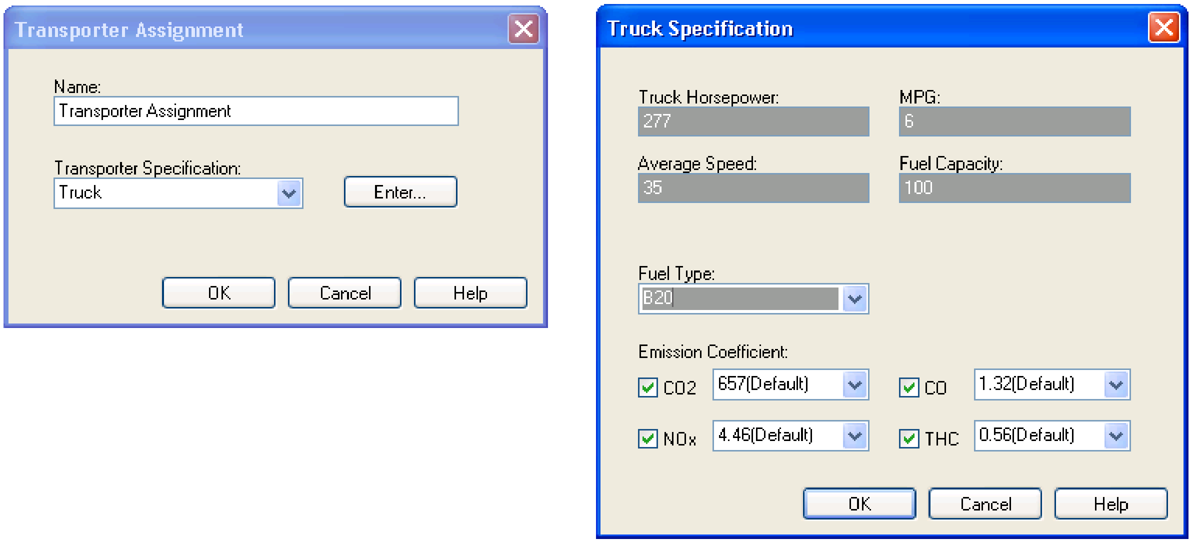
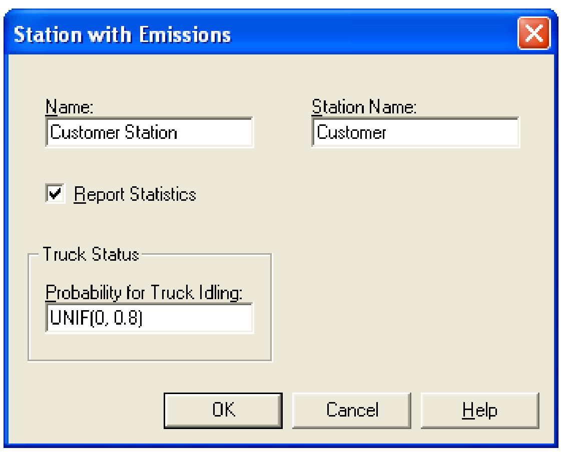
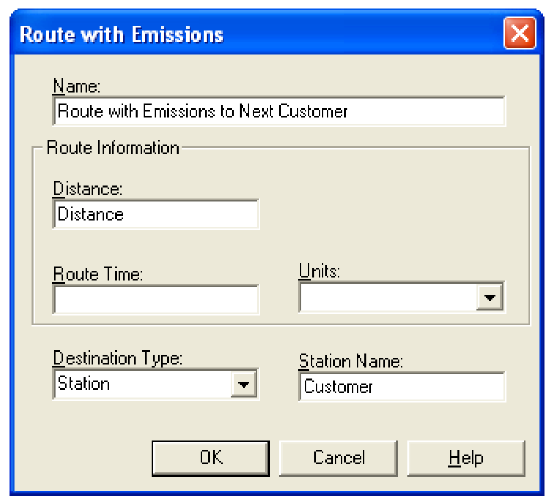


Figure 5 - 3: Transportation and Logistics Modules



(a)





(b) (c)

Figure 5 - 4: Simulation Module Interface: (a) *Truck Assignment* (b) *Route Station with Emissions*, (c) *Route with Emissions*

The number of customers served by each truck varies from truck to truck and from day to day. The travelling distance varies from truck to truck and from route to route as well. Table 5-3 shows an example instance of the route schedule for the eight trucks on a particular day.

Table 5 - 3: Generated Instance of the Route Schedule

Truck	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Route Distance	121.6	26.6	72.8	46.8	96.9	24.0	34.6	26.5
Number of Customers (Stops)	3	9	9	8	1	4	6	10

Additional information of the transportation system, including the truck information (e.g. horsepower, average speed, fuel type), routing assignment information (e.g. distance, number of customers), and delivery information (e.g. probability of truck idling, drop-off time), is assigned with given values. Table 5-4 shows the specification of each of the parameters used in the transportation system.

Table 5 - 4: Simulation Parameter Specification

Specification	**Assigned Value**
Number of Trucks	8 Trucks
Truck Horsepower	277 Horsepower
Truck Fuel Economy	6 Miles per Gallon
Truck Average Speed	35 Miles per Hour
Truck Fuel Capacity	100 Gallons
Fuel Type	20% Soya Blend Biodiesel
Route Distance	22 + 275×BETA(0.896, 2.77) Miles
Number of Customers	Random Integer between 1 and 10
Probability for Truck Idling	Random Number between 0 and 1
Number of Product Boxes Unloaded	Random Integer between 1 and 10
Unloading Time	2 Minutes per Box

With all the system parameters provided, a simulation model is built to represent the logistics and transportation system. Figure 5-5 shows the simulation model overview using standard modules. Eight trucks are created initially and their specifications are assigned to each truck. The truck, as the entity in the simulation model, moves to the assign module to have its routing information assigned, including the routing distance and the number of customers. The truck is then sent to the first customer using a route module. When the truck arrives at the first customer represented by a station module, the environmental performance measures are updated according to the mathematical relationship specified. The truck is determined to be left idle or turned off. After a certain delay time period for unloading the goods, the truck updates the environmental measures during the dropping-off process and checks for the next destination. If

the next destination is a customer, the truck is routed to that customer performing the same logic again. If the next destination is the warehouse, the truck is routed to the warehouse station, updates the environmental performance measures and collects other system performance measures including total time spent in the route, and total distance of the route.

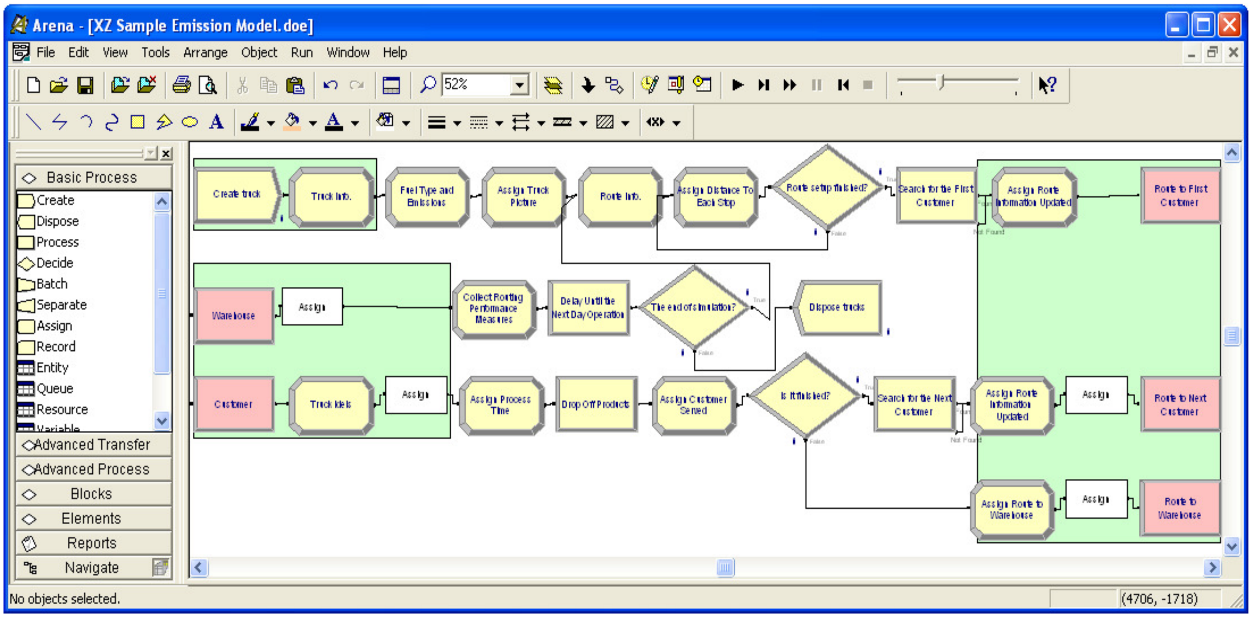


Figure 5 - 5: Example Simulation Model Using Standard Modules

With the model constructed using standard modules, the environmental performance measures, along with their implementation in the modules, have been analyzed to build a sustainability toolkit. The updates of the environmental performance measures happen and only happen at the occurrence of the arrival event and departure event. And the logic of the model behind the occurrence of these two events is unique regardless of which stop the truck arrives at or departs from. The following pieces of logic, which are also boxed in Figure 5-5, describe the generic sequences performed after the arrival and the departure event:

- Arrival Event: The truck updates the emissions generated from the location of previous customer; the truck is determined to be left idle or turned off.

- Departure Event: The truck updates the emissions generated during its unloading process in the location of the current customer; the truck searches for the next destination and routes to the next destination.

Thus, the nature that the system behaves as provides a more generic modeling idea of the simulation events. We introduce how the toolkit has been constructed in the following sections. In order to better illustrate the construction process, a flow chart of the logic and the pseudo codes are showed for each of the modules developed. These logic flows and pseudo codes are used to develop the toolkit despite what simulation language or software they are written in.

5.1.1.1 Truck Assignment Module

The *Truck Assignment* module is used to assign the general information for the vehicles selected (see Figure 5-6). Based on the vehicle type that is specified (e.g. Truck, Van, SUV), a second dialogue is opened for more detailed information about the vehicle. Taking the truck as an example, the user is asked to provide the information for the trucks, which includes the truck engine horsepower, the truck efficiency (measured in miles per gallon), the average speed of the truck and the fuel capacity. The user chooses the fuel type for the trucks to use. There are five types of fuel provided by our model, including the diesel, 10% soya blend biodiesel (B10), 20% soya blend biodiesel (B20), 30% soya blend biodiesel (B30) and 40% soya blend biodiesel (B40). Among four available emission components (CO, CO_2, NO_x, THC), the user may select which emission components they are interested in collecting.

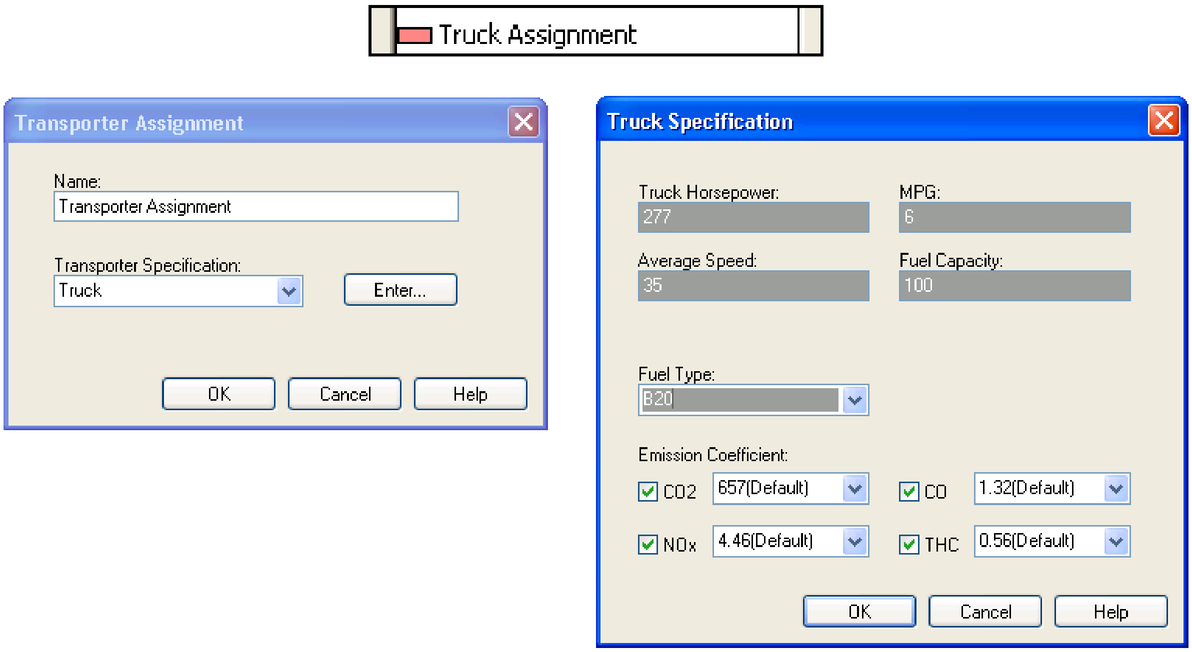


Figure 5 - 6: Transportation and Logistics Modules: *Truck Assignment* Module

Figure 5-7 is the logic flow for the *Truck Assignment* module. The truck information, including both the basic information and the environmental statistics related information, is specified first. In particular, the fuel type is chosen for running the truck. A decision is made to collect the corresponding environmental performance measure (or a set of them). If an environmental performance measure is chosen to be collected within the transportation model, the next step is to initialize the collection. Depending on target simulation package used, this step may include initial setups such us defining state variables and defining collection reports. The pseudo codes for carrying out the simulation logic are shown in Algorithm 5-1.

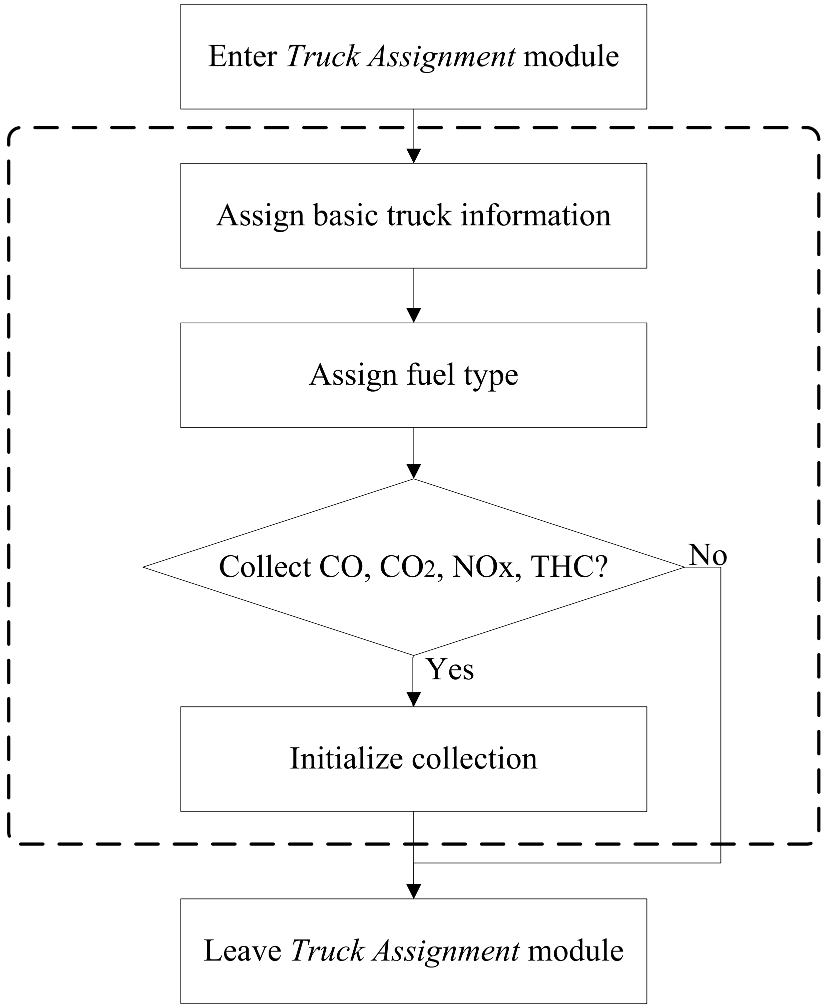


Figure 5 - 7: Flowchart of the logic for the *Truck Assignment* Module

Algorithm 5 - 1: Pseudo-code for the *Truck Assignment* Module

Step 1: Assign basic truck information;
Set Average Speed(s), Engine Horsepower (h), Truck Efficiency (e);
Step 2: Assign fuel type Z used;
Step 3: If CO, CO_2, NO_x, THC, or any of them is collected,
set $CE_Y = 0$ or 1, then go to Step 4;
If none of the emissions is collected,
set $CE_Y = 0$, go to Step 5;
Step 4: Setup corresponding state variables, including:
Set $f_Y = 0$;
Set $g_z = 0$;
Set $Idle = 0$ or 1;
Set t_0 = current simulation time;
Step 5: Continue simulation.

By checking the checkbox of emission coefficient in the module, the toolkit returns the amount of emissions emitted by each individual truck and the cumulative amount of emissions produced by all of the trucks in the system. The default mathematical relationship to calculate the emissions have been adapted from Manicom et al. (1993) listed in the previous section as Equation 5-1 and Equation 5-2:

$$f_Y(t) = C_{Y,Z}ht = \frac{C_{Y,Z}hd}{s}.$$

Manicom et al. (1993) provide a list of the emission coefficient $C_{Y,Z}$ which are used as the default value in the toolkit. The users have the control of changing these emission coefficient based on other studies that exist or may become available in the future, or from their own studies.

5.1.1.2 Route with Emissions Module

The *Route with Emissions* module is used to route the vehicle to its next destination (see Figure 5-8). The module specifies what the destination of the truck is (the destination station name). The module provides two options to the user for entering the route information. The user can either enter the route distance for the vehicle to travel or the routing time until the vehicle reaches its destination. The route with emissions module initials the simulation departure event to start collecting of the emissions generated during travelling. In addition, the module is able to calculate accumulating emissions that are generated while the truck remains at the current station. The system cumulative emissions are updated in this module as well.

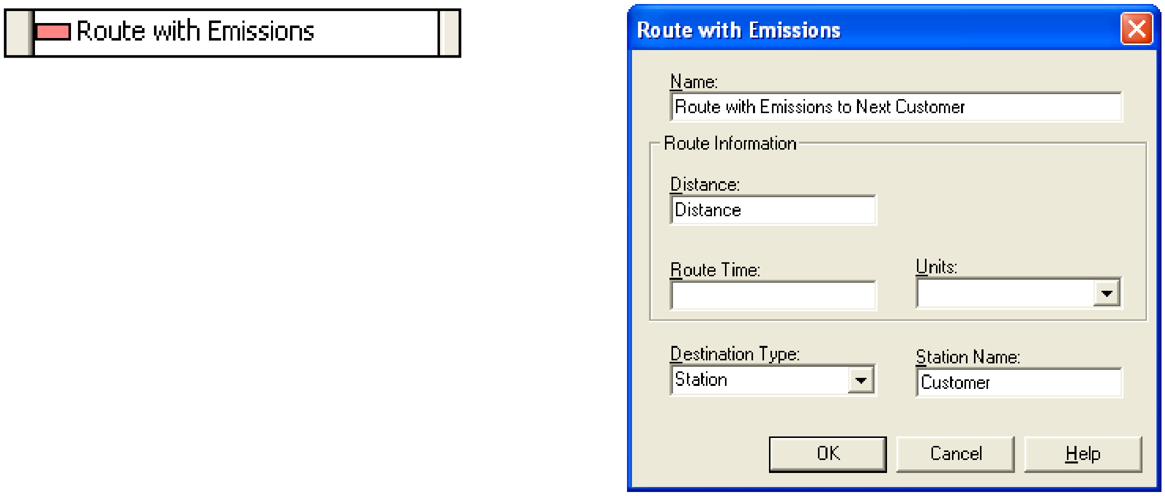


Figure 5 - 8: Transportation and Logistics Modules: *Route with Emissions* Module

Figure 5-9 depicts the logic flow for the *Route with Emissions* module. The module first assigns the routing information to the truck, including the destination, the routing time, or equally the routing distance to the next stop. Then the module evaluates the amount of emissions generated within the current location where the truck stays. Depending on the truck idling status, the mathematical calculations are carried out using the equations defined previously.

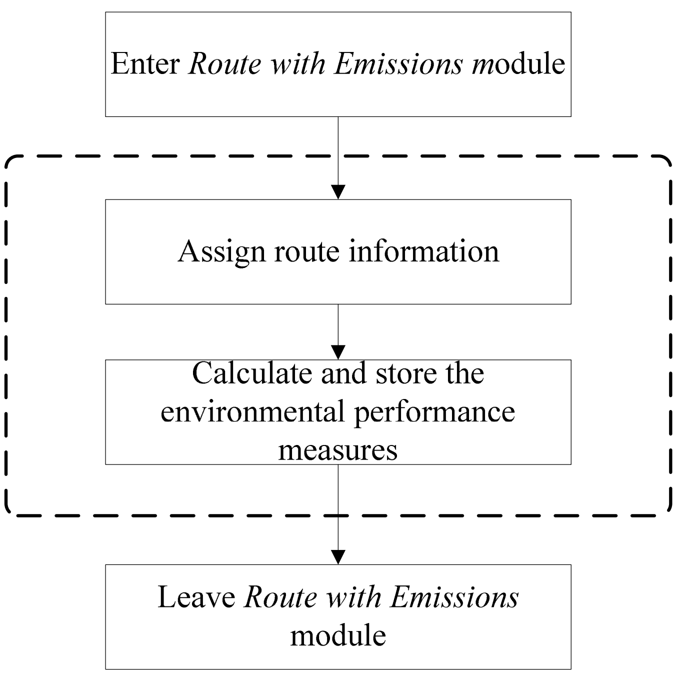


Figure 5 - 9: Flowchart of the logic for the *Route with Emissions* Module

The pseudo codes for executing the *Route with Emissions* module are shown in Algorithm 5-2.

Algorithm 5 - 2: Pseudo-code for the *Route with Emissions* Module

Step 1: Assign route information, specify the next stop S_i, the distance to the next stop d_{i-1}, or the traveling time t_{i-1};

Step 2: For each emission component Y, if $CE_Y = 1$, go to step 3;
For each emission component Y, if all $CE_Y = 0$, go to step 4;

Step 3: If $Idle = 1$, set t_1 = current simulation time;

Set $f_Y = C_{Y,Z} h(t_1 - t_0)$;

Set $E_Y = E_Y + f_Y$;

Set $TE_{Y,System} = TE_{Y,System} + f_Y$;

Step 4: Route the truck to the next stop.

5.1.1.3 Route Station with Emissions Module

The *Route Station with Emissions* module (see Figure 10) represents a physical location in simulation modeling. Additionally, when trucks arrive at a stop, which is modeled as a station, our *Route Station with Emissions* module collects the emissions generated along the route from the previous location. The module accumulates the system emissions as well. Finally, the user can specify the probability of the truck idling while it remains at the station location. The *Route Station with Emissions* also initializes the arrival event and triggers the collection of the emissions generated while the truck remains at the station, if the truck is left idling.

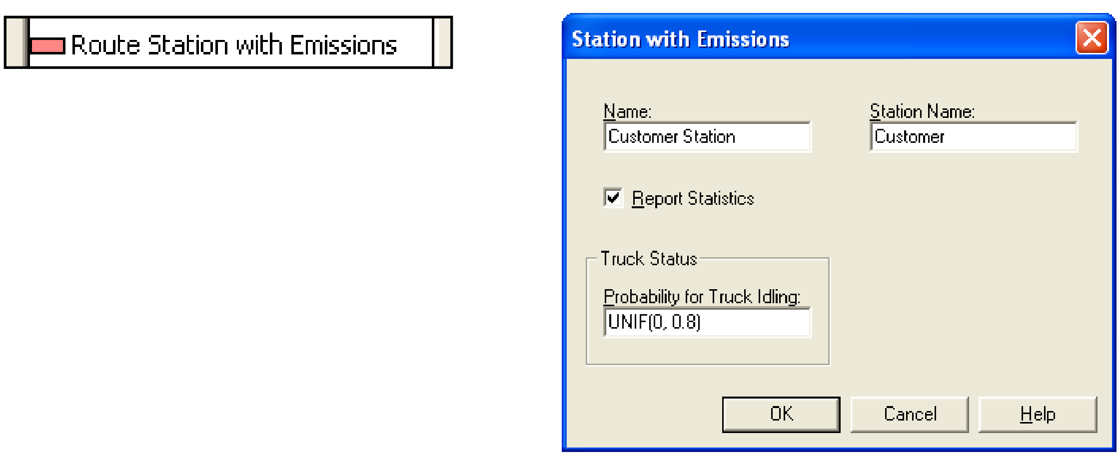


Figure 5 - 10: Transportation and Logistics Modules: *Route Station with Emissions* Module

Figure 5-11 shows the logic flow for the *Route Station with Emissions* module. When a truck arrives at a stop, which is represented by the *Route Station with Emissions* module, it is first determined to be left idling or turned off using certain probability (0 can stand for that the truck is turned off while 1 indicates that it is left idling). This setup belongs to assignments for the station information. Then the module captures the amount of emissions generated along the route from the previous stop. Again, the corresponding mathematical calculations are performed. The truck then continues with the simulation and actions such as unloading products are executed.

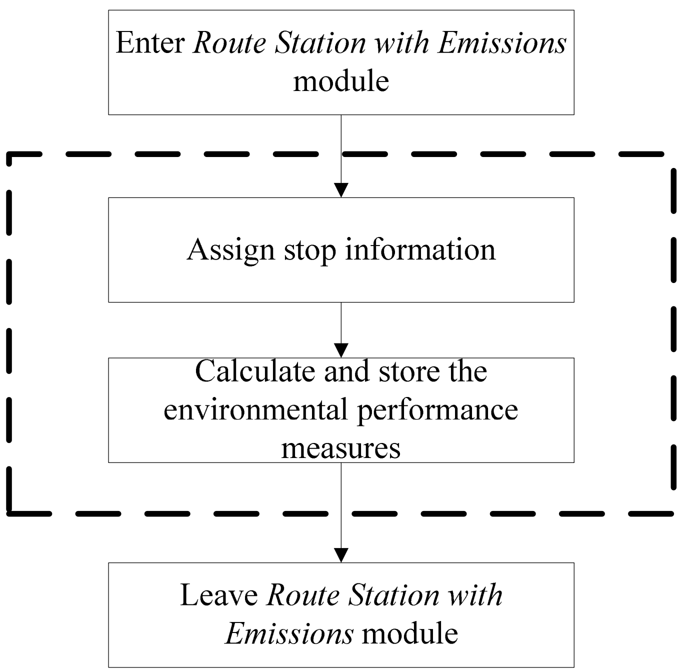


Figure 5 - 11: Flowchart of the logic for the *Route Station with Emissions* Module

The logic within the *Route with Emissions* module is programmed and carried out step by step. Algorithm 5-3 shows the pseudo codes for the *Route with Emissions* module.

Algorithm 5 - 3: Pseudo-code for the *Route Station with Emissions* Module

Step 1: In Stop i, sample truck idling probability;
Set $Idle = 0$ or 1;
Set t_0 = current simulation time;
Step 2: For each emission component Y, if $CE_Y = 1$, go to step 3;
For each emission component Y, if all $CE_Y = 0$, go to step 4;
Step 3: Based on the information obtained
Set $f_Y = C_{Y,Z} ht_{i-1}$ or $f_Y = C_{Y,Z} hd_{i-1} / s$;
Set $g_Z = d / e$;
Set $E_Y = E_Y + f_Y$;
Set $F_Z = F_Z + g_Z$;
Set $TE_{Y,System} = TE_{Y,System} + f_Y$;
Set $TF_{Z,System} = TF_{Z,System} + g_Z$;
Step 4: Continue simulation.

5.1.1.4 Remodel using the Transportation and Logistics Modules

To demonstrate the usage of the transportation and logistics modules that have been developed, the problem is remodeled using the three modules. All the system specifications and configurations remain the same as in the original model. Figure 5-12 displays the overview of the simulation model for using the modules along with standard modules to represent the transportation and logistics system.

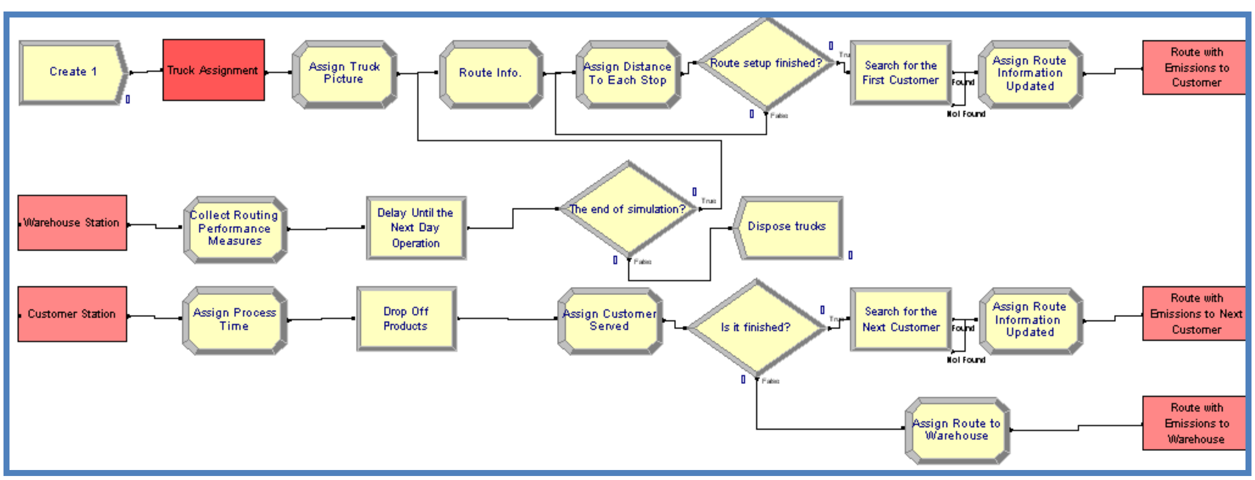


Figure 5-12: Simulation Model using Transportation and Logistics Modules

Referring to Figure 5-12, entities representing the trucks are created in the Create module. The *Truck Assignment* module is used to assign truck parameters as well as to specify the

emissions to collect. The routings for this example are specified in the traditional assign modules. The first *Route with Emissions* module (labeled Route with Emissions to Customer) is used to specify the routing of the truck from the warehouse to the first customer. The *Route Station with Emission* module (labeled Customer Station) represents the location of a "generic" customer. At each customer location, a Process module is used to represent the delivery process. The Decide module is used to determine if the route has been complete. If so, the *Route with Emissions* module (labeled Route with Emissions to Warehouse) is used to route the truck back to the warehouse. If the route is not complete, the *Route with Emissions* module (labeled Route with Emissions to Next Customer) is used to send the truck to the next customer on the route. The simulation model is set up to represent a simulation length of a 5 day week.

With the development and implementation of the sustainability toolkit, the model is set up and run for one period of five days simulation length. Table 5-5 shows the output results from running the model utilizing 20% soya blend biodiesel. The performance measures are collected and shown for each individual truck and for the entire system.

Table 5 - 5: Simulation Results for Trucks using 20% Soya Blend Biodiesel over 5 Days

	Truck 1	**Truck 2**	**Truck 3**	**Truck 4**	**Truck 5**	**Truck 6**	**Truck 7**	**Truck 8**	**System**
CO Emissions (gram)	8,276.2	2,650.5	5,442.2	5,021.6	4,728.0	5,877.8	5,724.4	6,286.3	44,007.0
CO_2 Emissions (gram)	4.1E+6	1.3E+6	2.7E+6	2.5E+6	2.4E+6	2.9E+6	2.8E+6	3.1E+6	2.2E+7
NO_x Emissions (gram)	2.8E+4	9.0E+3	1.8E+4	1.7E+4	1.6E+4	2.0E+4	1.9E+4	2.1E+4	1.5E+5
THC Emissions (gram)	3,511.1	1,124.4	2,308.8	2,130.3	2,005.8	2,493.6	2,428.5	2,666.9	18,669.0
Route Distance (mile)	703.6	198.9	401.9	381.5	384.9	492.6	482.6	532.9	3,579.0
Route Time (hour)	25.2	8.8	17.1	17.8	14.5	18.6	19.1	19.7	140.9
Fuel Consumed (gallon)	117.3	33.1	67.0	63.6	64.2	82.1	80.4	88.8	596.5
Customers Served	28.0	18.0	34.0	40.0	18.0	24.0	29.0	30.0	221.0

As shown from the results, the environmental performance measures are collected and reported as easily and readily as the traditional performance measures. To demonstrate the usage of the transportation and logistics toolkit in helping decision-making, a comparison for two alternative system configurations are conducted. The model is set up to compare the trucks using diesel fuel with trucks using 20% soya blend biodiesel. In order to compare the differences in the performance measures using these two types of fuel, the simulation model is run for 50 replications of one week (five days). The data has been used to construct 95% confidence intervals on the mean value for each performance measure. Table 5-6 shows the comparison of the results. From the result, it shows that using 20% Soya Blend Biodiesel can reduce the amount of carbon monoxide and total hydrocarbon significantly. However, it generates more nitrogen oxides, and doesn't have much effect in generating carbon dioxide.

Table 5 - 6: Comparison of 95% Confidence Intervals on System Performance Measures

	20% Soya Blend Biodiesel		**Diesel**	
	Average	**Half-width**	**Average**	**Half-width**
System CO Emissions	44,612.0	1,165.3	51,034.0	1,333.0
System CO_2 Emissions	2.2E+7	5.8E+5	2.2E+7	5.8E+5
System NO_x Emissions	1.5E+5	3.9E+3	1.9E+4	1.3E+3
System THC Emissions	18,926.0	494.4	24,334.0	635.6
System Route Distance	3,581.1	110.9	3,581.1	110.9
System Route Time	142.1	3.5	142.1	3.5
System Fuel Consumed	596.9	18.5	596.9	18.5

5.1.2 Verification, Validation and Testing of the Transportation and Logistics Modules

Verification of the developed transportation and logistics modules is to determine whether the sustainability toolkit performs as intended. The purpose of the emission toolkit developed for logistics and transportation systems is providing the users with the ability to use the template directly with other standard modules while collecting the emissions generated behind the scene. To verify that the model using emission toolkit does result in the same answers

as that provided by the original model without the toolkit, these two models are run for 100 replications of one week (5 days) period. A 95% confidence interval of the mean value for each performance measure has been constructed. Because the seeds used to generate random numbers can vary when running different models, these two models are not resulting in the exact same answers for each replication. However, during 100 replications for both models, the average system emission values are reasonably the same. Two-sample-t test also concludes that no evidence of statistical differences is found on the mean values resulted from the two models. The two-sample-t test is performed using Arena Output and the results are shown in Table 5-7. These two models are verified to provide the same results and the emission toolkit is verified and tested to achieve the desired purpose.

Table 5 - 7: Toolkit Testing Results of the Transportation and Logistics Modules

	Original Model without Toolkit	**Model with Toolkit**	**Value Difference**	
	Average	**Average**	**95% CI Min Value**	**95% CI Max Value**
System CO Emissions	44,633	44,645	-1110	1090
System CO_2 Emissions	2.2E+7	2.2E+7	-5.5E+5	5.4E+5
System NO_x Emissions	1.5E+5	1.5E+5	-3750	3670
System THC Emissions	18,935	18,940	-471	461

Validation of the sustainability toolkit is primarily focused on whether the simulation model using the Emission Toolkit template is an accurate representation of the studied paper. The results are validated using the information provided by Manicom et al. (1993). For instance, in Manicom's paper, the change of THC by using B20 from the baseline diesel is -22.2%. In the prototype transportation problem using simulation-based emissions toolkit, this number is reflected as -22.2% as well.

5.1.3 Capabilities and Limitations of the Transportation and Logistics Modules

The designed transportation and logistics modules show the benefits that can be provided to the user interested in including environmental impacts performance measures as an integral part of the decision making process for transportation and logistics problems. The modules have the ability to collect and report the environmental performance measures, including the fuel consumption and emissions generation for CO, CO_2, NO_x, and THC. These performance measures are shown to the user either by different running vehicles or by the entire simulation system. The transportation and logistics modules can be used along with other standard modules in a drag-and-drop fashion easily and do not require additional repeatable manual inputs of the mathematical and logic relationship for calculating the desired environmental performance measures. The modules are developed and can be used in a very efficient manner. The inputs are only limited to vehicle information including the basic and environmental impacts relation information, which is flexible for various logistics and transportation systems. The set of modules is compact in a high level of model programming. It provides the capability of modeling complex transportation systems and obtaining the environmental impact measures with least modules needed (three modules). At the end of the simulation, a report containing all the selected environmental performance measures is displayed.

Some of the limitations of the transportation and logistics modules are listed as follows:

- The mathematical relationship used does not take into account the truck load status, and the different speed the truck is travelling at.
- The collected environmental performance measures are limited in a way that only five different types of fuel can be used including baseline diesel, B10, B20, B30 and B40, while only four types of emission can be evaluated including CO, CO_2, NO_x, and THC.

Some other typical environmental factors in transportation systems such as particulate matter (PM) are not considered in the modules developed.

- The emissions generation and fuel consumption occur constantly over time. However, the performance measures are collected discretely in the modules; and
- The modules do not reflect the differences between a loaded truck and an empty truck.

5.1.4 Summary of the Transportation and Logistics Modules

In this section we have demonstrated the development of the sustainability toolkit for transportation and logistics systems. The toolkit development methodology is applied to design the transportation and logistics modules framework. In order to demonstrate the utilization of the toolkit, a test example is introduced. Logic flows and the pseudo codes for the transportation and logistics modules are developed and then implemented using ARENA simulation software. Simulation models are generated for the test example for demonstration and toolkit verification and testing.

Some further work that can be done to the currently developed transportation and logistics modules will be focused on the following areas:

- Obtain from more recent literature for up-do-date data to calculate desired environmental performance measures and also include commonly studied emissions such as particulate matter;
- Obtain from other research and study for mathematical and statistical relationships where truck loaded factor and running speed are factors in determining the emissions generation and fuel consumption;
- Develop a more general mechanism where the users can define their own preferred mathematical equations to collect their interested environmental performance measures.

The General Sustainability Toolkit that will be introduced later in the paper is one possible solution for this purpose.

5.2 Process with Emissions Module

Industrial and manufacturing processes often involve sustainability issues such as energy usage, raw material consumption and emissions and wastes generation. To capture the environmental performance measures within industrial processes, a *Process with Emissions* module has been developed. The module provides the user with a flexible framework to model industrial processes including industrial coating processes, injection molding processes and plastics processing. The development methodology has been applied first. Following the same developing steps, we identified the sustainable factors, the environmental performance measures, the state variables, and the system events that update the system state variables. The mathematical relationships are established. The industrial processes module is implemented and tested for verification and validation. An example of an industrial coating process is introduced for demonstration and illustration purposes. The system components of injection molding process and plastics processing are later discussed in this section as well.

Example 5-2: *A plant has a top coating line for coating vehicles. The top coating process consists of five sequential steps including underbody protection coating, filler application, base coat application, clear coat application and cavity sealing. In each of the five coating processes, the company can choose from solvent-based paint or water-based paint as the coating material. The company wants to evaluate this coating line on its productivity and efficiency, while taking into account the material consumption and the emissions generation, especially the generation of Volatile Organic Compounds (VOC). Figure 5-13 displays a flow chart of the stated system.*

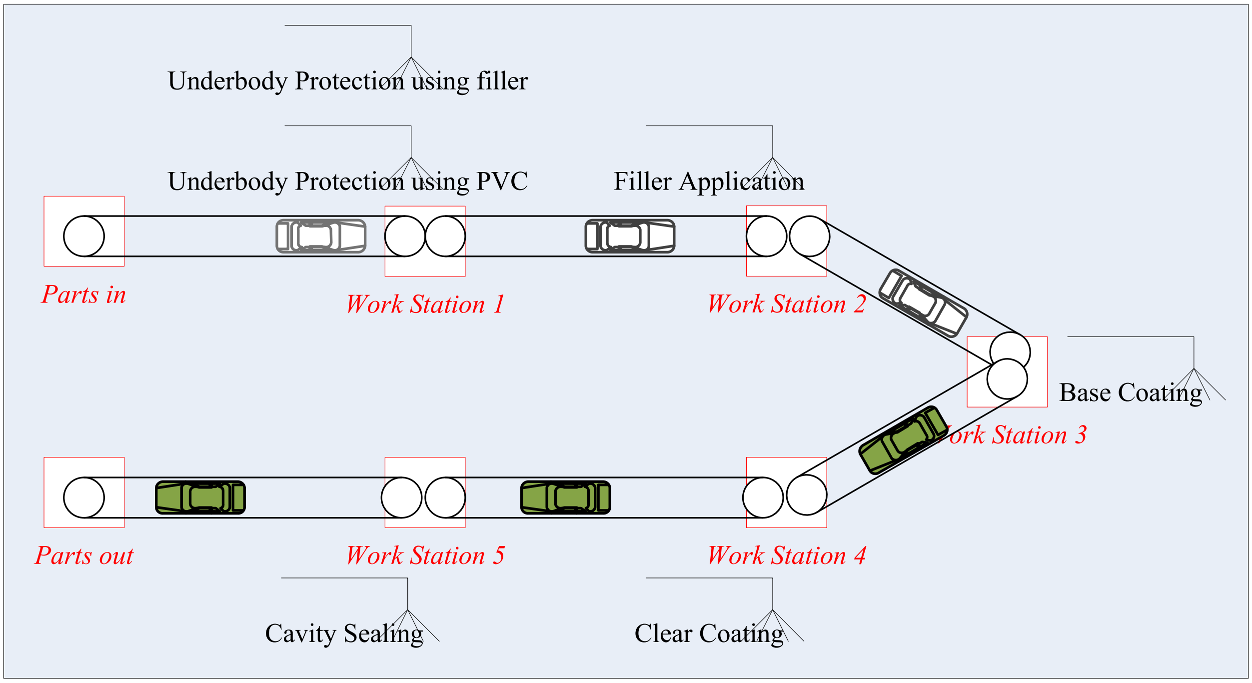


Figure 5 - 13: Flow Chart of the Industrial Coating System

The example shows a typical industrial coating process where environmental impacts become an important integrated part of decision-making. The effort is made to design and develop a flexible framework where predefined processes and statistics can be easily utilized while the users are also allowed to collect additional statistics by themselves. The toolkit development methodology is applied step by step.

Identify the environmental performance measures and define the simulation state variable. Some assumptions are made to develop the modules including:

1. The material consumption and emissions generation are not taking into consideration of the processing time;
2. The water-based paint consumption is not taking into consideration of the solvent ratio of the paint. Also the solvent-based paint consumption is not taking into consideration of the solvent ratio of the particular paint used.

The environmental performance measures of industrial and manufacturing processes typically fall into two main categories including energy and material consumption, and emissions and wastes generation. The industrial coating process involves in the material consumption such as paint and solvent usage. Volatile organic compound (VOC) generation is identified as another environmental performance measure. A major source of man-made VOCs is solvents, especially paints and protective coatings (WiKi). The VOC generation needs to be evaluated in the industrial coating process. The system state variables are defined quantitatively to represent the material consumption and the emission generation. Taking Example 5-2 in particular, the paint consumption state variable is defined to be $P_{Z,\, j}$, as paint consumed in process *j* by using paint type *Z*, solvent consumption state variable is $S_{Z,\, j}$, as solvent consumed in process *j* by using paint type *Z*. VOC generation consumption state variable is $V_{Z,\, j}$, as the volatile organic compounds generated in process *j* by using paint type *Z*. $P_{Z,\, j}$, $S_{Z,\, j}$, and $V_{Z,\, j}$ are process-based state variables. Paint consumption in the entire system (TP_{System}), solvent consumption in the entire system (TS_{System}) and VOC generated with the sytem (TV_{System}) are defined as the system level state variables to capture the environmental performance measures for the system.

Determine the simulation events. The industrial coating to vehicles process starts to consume coating materials, and generate emissions, once the process starts. Similarly, the coating process stops consuming paint and solvent, and stops generating VOC, when the process itself stops. Therefore, the simulation event which triggers the collection of environmental performance measures is the entity vehicle's entrance to spray machine/worker service. And the event which triggers the end collection of the measures is the entity vehicle's departure from the spray machine/worker.

Formulate the mathematical relationship. The predefined industrial coating process parameters are based on a study conducted by Rentz et al. (2002). The quantity of paint consumed to coat the vehicles is a function of several parameters including the coating area, the paint usage coefficient, and the coating process efficiency. Rentz et al. (2002) provide a list of the paint usage coefficient for various coating material including water-based paint and solvent-based paint. The paint usage coefficient is in the unit of grams per square meter (grams/m^2). The equation for calculating the paint consumption is:

$$f_j(x, Z) = \frac{C_Z A(x)}{eff}$$

where the paint consumed *f* by using paint type *Z* to paint vehicle *x* in process *j* is equal to the paint usage coefficient C_Z for paint type *Z* times the painting area *A(x)* for vehicle *x* divided by the painting process efficiency *eff*. The solvent consumption is calculated as:

$$g_j(x, Z) = R_Z f(x, Z)$$

where the solvent consumed *g* by using paint type *z* to paint vehicle *x* in process *j* equal to the paint consumption *f* times the solvent ratio of paint type *Z*, which is R_Z. Rentz et al. (2002) study the emissions generated from coating the vehicle as well. The results of the research study provide that the volatile organic compounds (VOC) generated *V* by using paint type *Z* to paint vehicle *x* in process *j* is calculated as the VOC emission coefficient of paint type *Z*, defined as τ_Z, times the painting area *A(x)* for vehicle *x* divided by the painting process efficiency *eff* resulting in the following relationship:

$$V_j(x, Z) = \frac{\tau_Z A(x)}{eff}.$$

Table 5-8 summarizes the notations and definitions of the state variables and parameters used in the industrial coating process.

Table 5 - 8: Parameters and State Variables for Industrial Coating Process

Notation	Definition	Unit
Z	Paint Type	---
x	A specific Vehicle	---
j	A specific Process	---
$P_{Z,j}$	Paint Consumption at Process *j*	grams
$S_{Z,j}$	Solvent Consumption at Process *j*	grams
$V_{Z,j}$	VOC Generation at Process *j*	grams
TP_{System}	System Paint Consumption	grams
TS_{System}	System Solvent Consumption	grams
TV_{System}	System VOC Generation	grams
A	Painted Area	m^2, $feet^2$, $inch^2$
eff	Coating Process Efficiency	percentage
C_Z	Paint Usage Coefficient for Paint Type Z	$grams/m^2$
R_Z	Solvent Ratio of Paint Type Z	percentage
τ_Z	VOC Emission Coefficient for Paint Type Z	$grams/m^2$

In a simulation model consisting of processes, traditional system performance measures may include the average process resource utilization, average process time, and average waiting time. The traditional performance measures are built-in statistics in the target simulation package.

5.2.1 Process with Emissions Module Development

Once the state variables, events and the mathematical relationship are defined, the sustainability for industrial coating process is constructed and implemented.

Figure 5-14 shows the *Process with Emissions* module panel from Figure 5-1 sustainability toolkit developed. This module is designed to model and return sustainability measures for system components including industrial coating process, injection molding process and plastics processing process. Figure 5-15 shows the user interface of the *Process with Emissions* module. The upper portion of the dialog box provides the functions of a traditional process module. The user specifies the process resources and processing time.

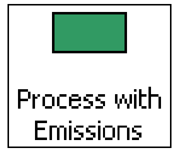


Figure 5-14: Process with Emissions Module

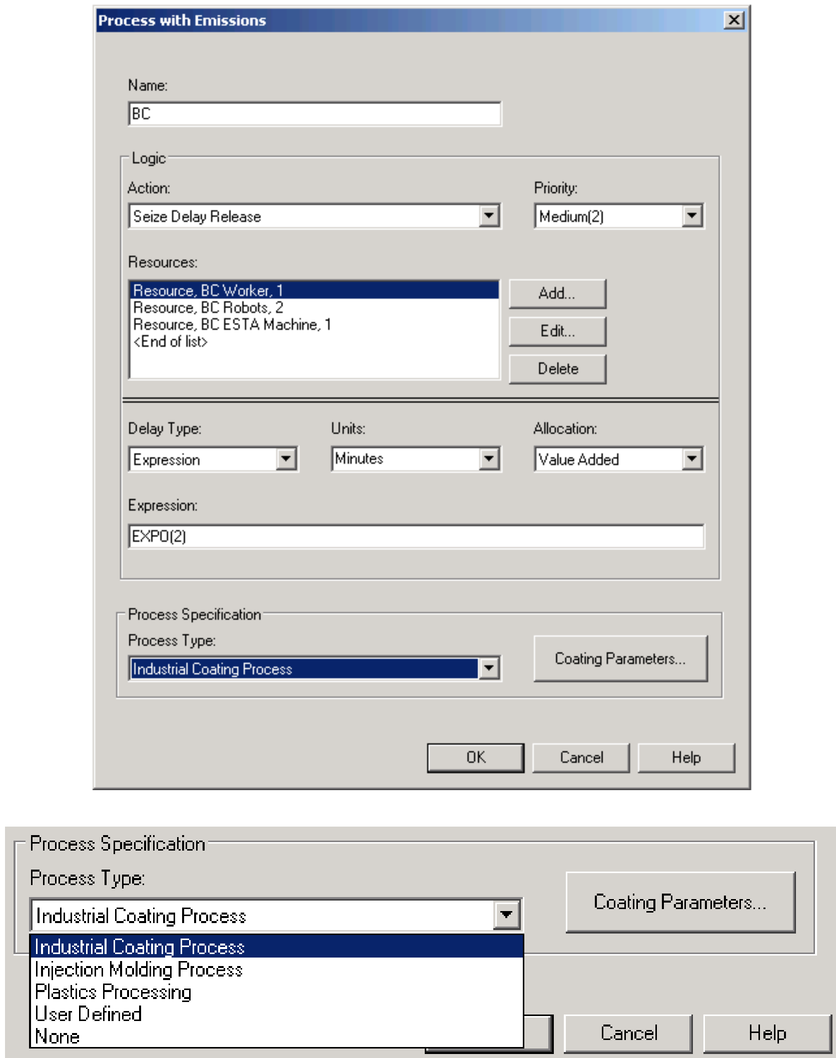


Figure 5 - 15: Process with Emission Dialog Box

In the lower portion of the *Process with Emissions* module, additional functionality is provided for process specifications. The industrial coating process, injection molding process and the plastics processing process are shown in the drop-down menu. The 'user defined' option is where the users can specify their own process parameters and define their own environmental

statistics to collect. For a standard industrial coating process, in Figure 5-16, system parameters are entered by the user. The user needs to choose a paint type from water-based paint and solvent-based paint. Some commonly used solvent ratios are provided to the user or the use can define the solvent ratio. Surface area also needs to be specified with given area unit. The user can enter any expression for the coating efficiency parameter. Lastly, the user can check the checkbox for collecting and reporting the interested environment measures. Details of the coating process dialog are discussed in later sections.

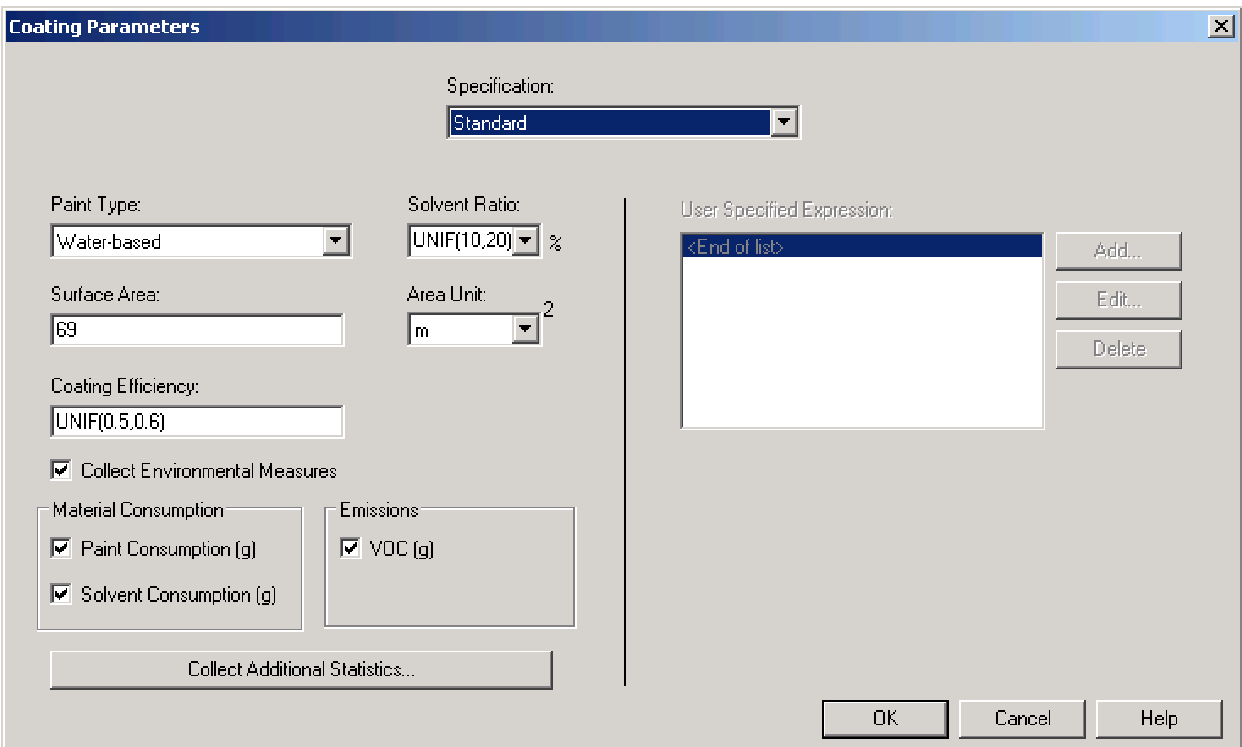


Figure 5 - 16: Standard Industrial Coating Process Parameters

5.2.1.1 Simulation Model Construction using Standard Simulation Modules

To better illustrate the development steps and the benefits of the module, specific system configurations have been added to example 5.2. Table 5-9 shows the detailed material selection

for each of the coating processes including the paint type and the corresponding solvent ratio of the paint used.

Table 5 - 9: Material Used for Coating Processes Example 5-2

	Underbody Protection using filler	**Underbody Protection using PVC**	**Filler Application**	**Base Coating**	**Clear Coating**	**Cavity Sealing**
Paint Type	Water-based	PVC	Water-based	Water-based	Solvent-based	Powder
Solvent Ratio	7%	-	7%	UNIF(10%,20%)	40%	-

In this particular system, there are three different types of coating conditions. The Underbody Protection using filler, the Filler Application, and the Clear Coating processes use the pre-defined paint type and collect paint consumption, solvent consumption and VOC generation using the pre-defined mathematical relationship. These processes are referred to as the standard processes. In some cases, such as Base Coating process, the painting material is chosen, default environmental performance measures are collected. In addition to these default measures, the process requires to collect CO, NOx, and dust emissions. These processes are referred to as the standard processes plus additional user-defined statistics. The PVC and Cavity Sealing processes do not use paint as the coating material, therefore are not collecting default environmental statistics as well. These processes are pure user-defined processes. Table 5-10 shows these three categories of process types and they can all be modeled using the *Process with Emissions* module.

Table 5 - 10: Process Types Definition

Process Types	**Specification**
Standard Process	Pre-defined environmental statistics are collected using pre-defined process parameters
Standard Process plus additional user-specified statistics	Standard process with some additional statistics defined and collected by user
User-defined Process	User has full control of defining and collecting environmental statistics

With the system configuration defined, a simulation model is constructed using standard modules (see Figure 5-17). The entities, which are the vehicles to be coated, flow through each of the processes. In the boxed areas, the collection of the environmental performance measures are performed and stored as the vehicles going through the processes and being painted.

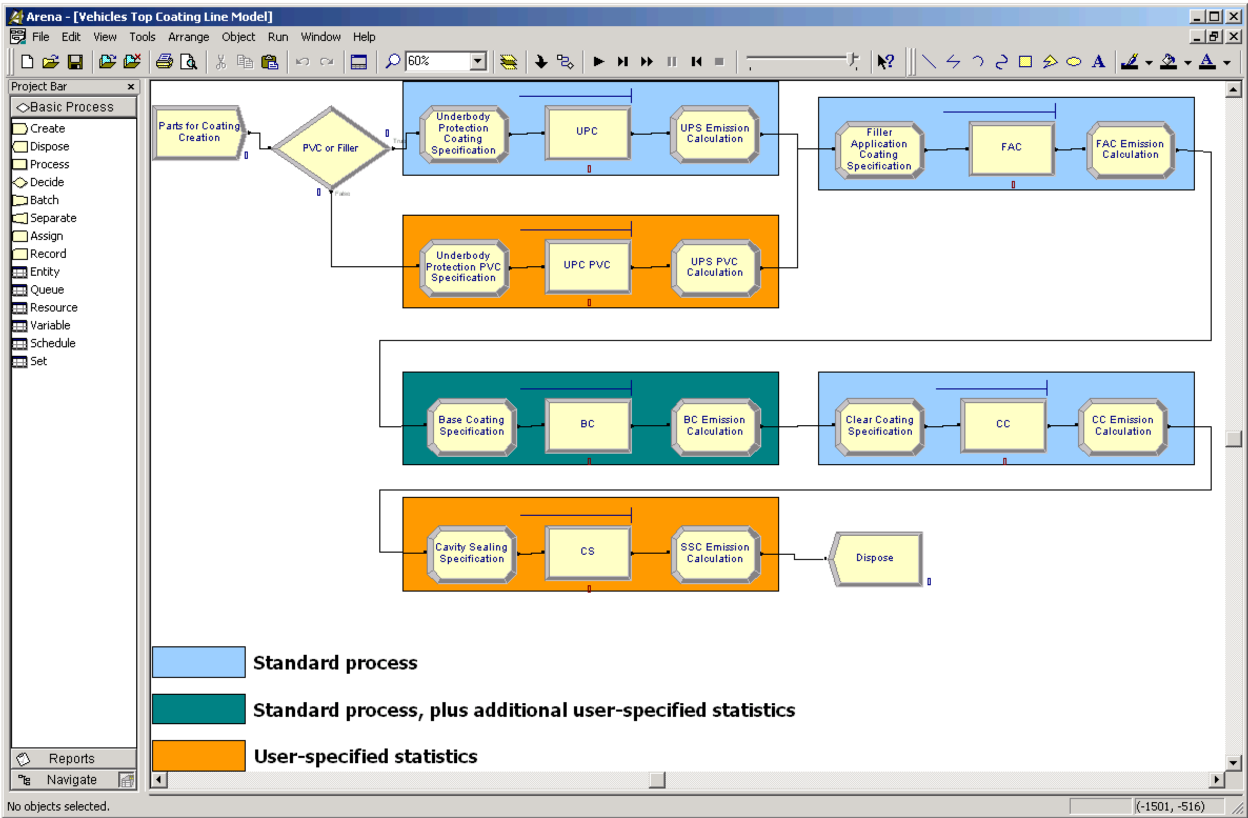


Figure 5 - 17: Coating Processes Simulation Model using Standard Modules

5.2.1.2 Logic Flow and Pseudo Code for the Process with Emissions Module

The boxed areas in Figure 5-17 are designed into the *Process with Emissions* module. Since the areas follows the same logic flow (shown in 5-18), they can be grouped together as an integrated logic flow in the *Process with Emissions* module. The module will first assign the basic process information including action type, resources, and processing time. Depending on the process type, a standard process will require the user to input coating parameters showing in

Figure 5-16. If the user has additional statistics besides the standard ones, the module will ask the user to specify their own statistics. A user-defined process requires inputs of user specified statistics as well. All performance measures are calculated and stored within the *Process with Emissions* module. Algorithm 5-4 shows the pseudo codes for carrying out the logic flow.

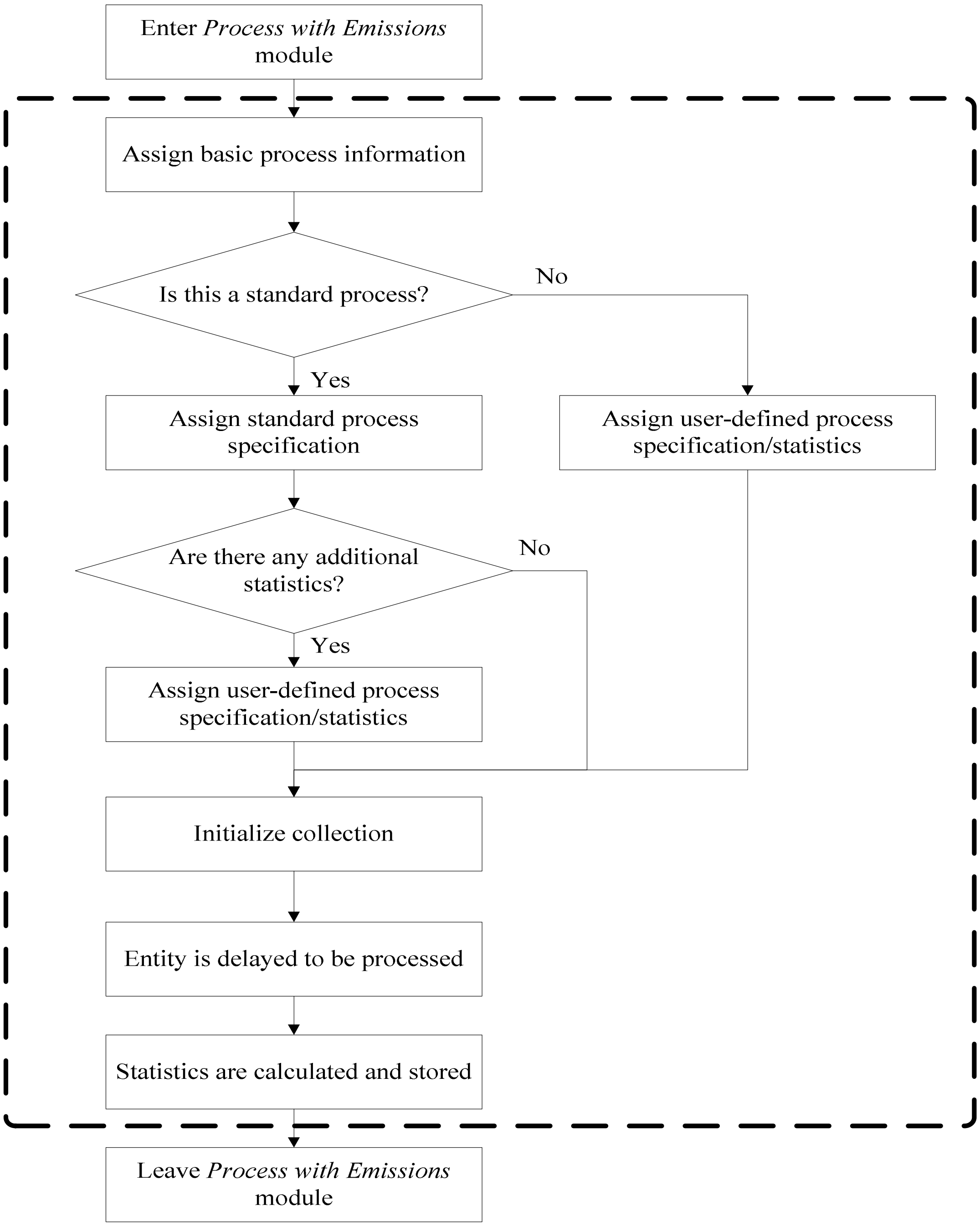


Figure 5 - 18: Flowchart of the logic for the *Process with Emissions* module

Algorithm 5 - 4: Pseudo-code for the *Process with Emissions* module

Step 1: Assign basic process information including:
action type, resources, and processing time;

Step 2: If the process is a standard process, go to Step 3;
If the process is a user-defined process, go to Step 5;

Step 3: Assign paint type Z, solvent ratio R_Z, painted surface area A,
and coating efficiency *eff*;

Step 4: If there is additional user-defined statistics to collect, go to Step 5;
If no additional user-defined statistics, go to Step 6;

Step 5: Assign initial values for collecting user-defined statistics;

Step 6: Delay entity for an amount of processing time;

Step 7: If the process is a standard process, go to Step 8;
If the process is a user-defined process, go to Step 10;

Step 8: Set $f_j(Z) = \dfrac{C_Z A}{eff}$, $g_j(Z) = R_Z f_j(Z)$, $V_j(Z) = \dfrac{\tau_Z A}{eff}$;

$P_{Z,j} = P_{Z,j} + f_j(Z)$, $S_{Z,j} = S_{Z,j} + g_j(Z)$, $V_{Z,j} = V_{Z,j} + V_j(Z)$;
$TP_{System} = TP_{System} + f_j(Z)$, $TS_{System} = TS_{System} + g_j(Z)$;
$TV_{System} = TV_{System} + V_j(Z)$;

Step 9: If there is additional user-defined statistics to collect, go to Step 10;
If no additional user-defined statistics, go to Step 11;

Step 10: Collect and store user-defined statistics;

Step 11: Continue simulation.

5.2.1.3 Model using the Process with Emissions Module

To demonstrate the usage of the *Process with Emissions* module, the coating system has been remodeled using the module. All system specifications and configurations are the same as in the original model. Figure 5-19 is an overview of the simulation model using the *Process with Emissions* module, along with standard modules, to represent the industrial coating system. Now the entity vehicle flows through each of the coating processes. Instead having three standard modules and inputting all the calculations for each one of them, the user only needs one process with emissions module and enters several coating parameters. The model is set up and run to illustrate the module usage. Table 5-11 shows the output of the simulation model run for one replication over 20 hours. The sustainability performance measures are collected and shown for each process and for the entire system.

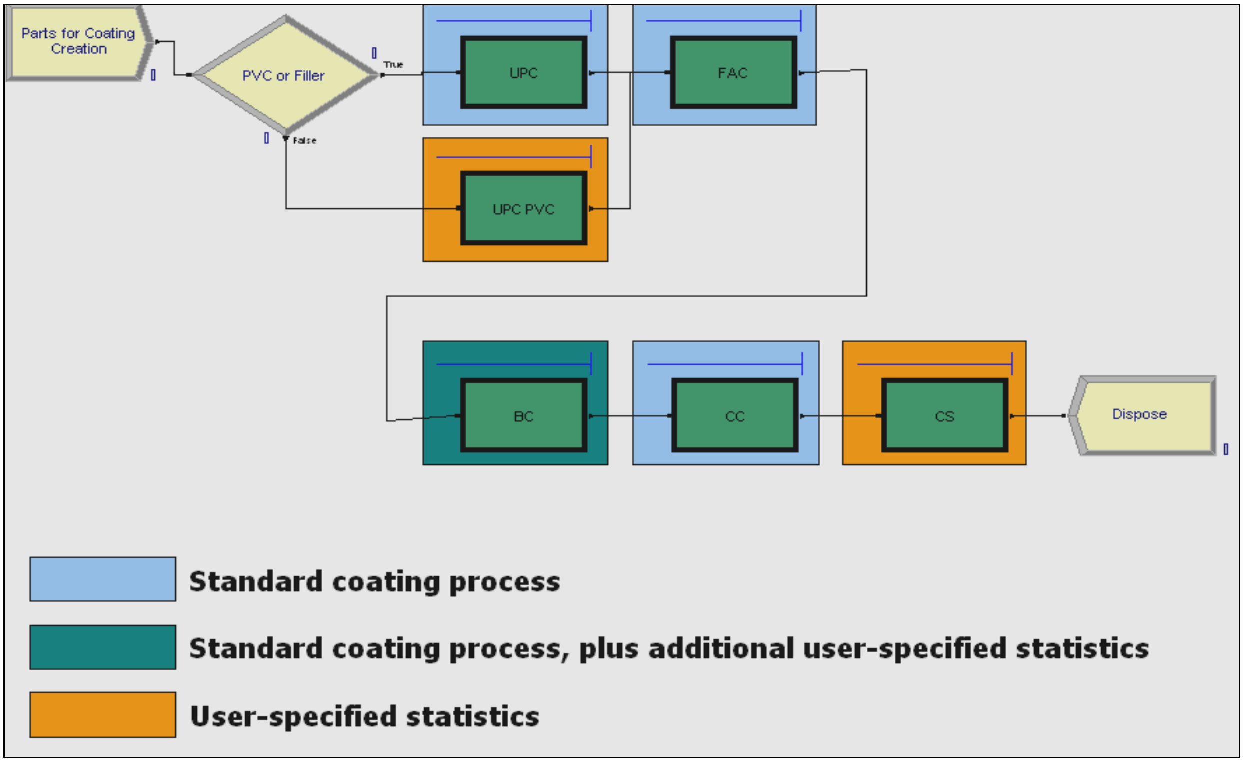


Figure 5 - 19: Simulation Model using *Process with Emissions* Modules

Table 5-11: Simulation Results for Vehicles Top Coating Line over 20 hours

	Underbody Protection using filler	**Underbody Protection using PVC**	**Filler Application**	**Base Coating**	**Clear Coating**	**Cavity Sealing**	**System**
Consumption							
Paint Consumption (g)	2.1E+05	-	2.3E+06	2.3E+06	1.3E+06	-	6.4E+06
Solvent Consumption (g)	1.5E+04	-	1.6E+05	3.6E+05	5.2E+05	-	1.1E+06
Add. PVC Usage (g)	-	3.5E+03	-	-	-	-	-
Add. Powder Usage (g)	-	-	-	-	-	5.5E+03	-
Emissions							
VOC Generation (g)	8.1E+04	-	8.9E+05	8.9E+05	2.6E+06	-	4.5E+06
Add. CO Generation (g)	-	-	-	4.8E+03	-	-	4.8E+03
Add. NOx Generation(g)	-	-	-	1.8E+04	-	-	1.8E+04
Add. Dust Generation(g)	-	-	-	2.3E+03	-	-	2.3E+03

5.2.2 Verification, Validation and Testing of the Process with Emissions Module

Verification of the *Process with Emission* module is to examine the toolkit functionalities and determine whether it functions as desired. The simulation model using the module has been debugged. No errors or warnings are shown running the model with the *Process with Emission* module. The simulation model using the module returns the desired statistics, namely the paint consumption, the solvent consumption and the VOC generation, for each process and for the entire system. Additional user-defined statistics are able to be collected and shown at the end of the report. An example running output using the module is shown in the previous section. Testing of the *Process with Emission* module is to examine that the module representing the system will not have an effect on the system itself. The module's main function is to collection environmental performance measures in the system. Running 100 replications, table 5-12 shows a module testing of 95% confidence interval on the main environmental performance measures for the original model without using the toolkit and for the simulation model using the toolkit. Compared with each other, the model with the toolkit provides reasonably the same results. In order to prove it statistically, a two-sample-t test is running to compare the system level performance measures means using Arena Output Analyzer. The results are also shown in Table 5-12. Since value of zero is included in the 95% confidence interval, there are no statistical differences found between the performance measures obtained from the original model and the model using the toolkit.

Table 5 - 12: Toolkit Testing Results of the Process with Emissions Module

		Original Model without Toolkit	Model with Toolkit	Difference Value	
		Average	Average	95% CI Min Value	95% CI Max Value
Paint Consumption	UPC	2.53E+05	2.52E+05	-9680	10300
	FAC	2.44E+06	2.45E+06	-31100	15200
	BC	2.38E+06	2.38E+06	-21200	22600
	CC	1.33E+06	1.33E+06	-9510	12700
	System	6.40E+06	6.41E+06	-5.97E+04	4.90E+04
Solvent Consumption	UPC	17684	17663	-678	721
	FAC	1.71E+05	1.71E+05	-2180	1060
	BC	3.57E+05	3.58E+05	-15400	12900
	CC	5.31E+05	5.31E+05	-3800	5090
	System	1.08E+06	1.08E+06	-8670	8930
VOC Generation	UPC	96553	96330	-3620	4070
	FAC	9.30E+05	9.33E+05	-12700	5720
	BC	9.07E+05	9.06E+05	-7760	9200
	CC	2.66E+06	2.66E+06	-19900	24900
	System	4.59E+06	4.59E+06	-37000	37000

5.2.3 Process with Emissions Module Component for Injection Molding Process

Injection molding processes are commonly used manufacturing processes today. It is used to produce parts from thermoplastic or thermosetting plastic materials. Injection molding process mainly consists of melting the polymer resin, mixing with additives, injecting into a mold cavity, and cooling by air or water. The plastic is hardened to the configuration of the mold cavity. Thiriez (2006) shows that the overall injection molding energy consumption (excluding polymer production) in the U.S. on a yearly basis amounts to 2.06×10^{8} GJ. His paper also studies the environmental impacts of injection molding processes among three different types of injection molding machines, including hydraulic machines, all-electric machines and hybrid machines.

Figure 5-20 shows the interface of the injection molding process component of the *Process with Emissions* module. For a standard injection molding process, which means the module will return the default built-in environmental performance measures, the user will choose a type of machine to use, indicate the input polymer weight, and then choose the environmental performance measures to collect, which include energy consumption and four types of air pollution generation.

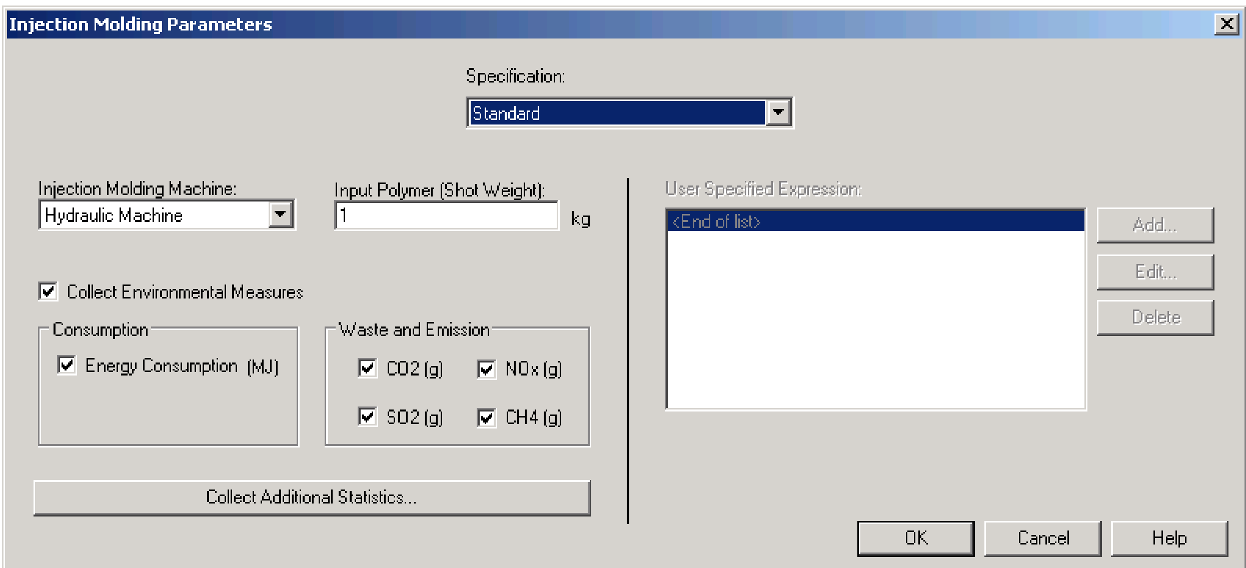


Figure 5 - 20: Injection Molding Process Module Interface

Some assumptions are made when using the toolkit module for injection molding process including:

1. Plastic types are not specified in this module component. It is assumed that the environmental performance measures collected are the average energy consumption and average emissions generation by using a particular machine type.
2. The environmental performance measures are collected for the injection molding stage only. Compounder stage emissions, polymer production emissions and processing emissions are not included in the module component.

3. It is assumed that the injection molding process does not take into consideration of the process efficiency.

The environmental performance measures for injection molding process include the energy consumption and the emissions generation in this research. Corresponding simulation state variable are defined. Thus, there are two sets of the state variables, one set represents the energy consumption and the other set of state variables represents the emissions generation. $R_{M,j}$ is defined as the state variable of the specific energy consumption at process j by using injection molding machine M. Similarly, $E_{Y,j}$ is the state variable representing the amount of emission component Y generated in injection molding process j. TR_{System}, $TE_{Y, System}$ are the two state variables taking from the system point of view where TR_{System} represents the state variable of the specific energy consumption for the entire system, and $TE_{Y, System}$ is the state variable for emission component Y generation within the system. Table 5-13 summarizes the state variables and some system parameter notations used for the injection molding process. In Thiriez's paper, the energy consumption, as well as four types of emissions including carbon dioxide (CO_2), sulfur dioxide (SO_2), nitrogen oxides (NOx) and methane (CH_4), are studied for all three types of injection molding machines. The energy consumption (f) from using injection machine M, and the emission generation (g), are both functions of the input polymer weight (m_j). Thiriez's paper provides a list of energy coefficients (CR_M), in kilo joule per kilo gram, for all three types of machines. The paper also provides a list of emission coefficients ($CE_{M, Y}$), in grams per kilo gram, for each emission component using three types of machines. The mathematical formulation for specific energy consumption and emissions generation are specified as

$$f_M(m_j) = m_j \cdot CR_M$$

$$g_Y(m_j) = m_j \cdot CE_{M,Y}.$$

Table 5 - 13: Parameters and State Variables for Injection Molding Process

Notation	Definition	Unit
M	Injection Molding Machine Type	---
j	A specific Process	---
Y	Emission Component	---
m_j	Amount of Plastic Processed in Process j	kg
CR_M	Energy Coefficient for using Machine M	MJ/kg
$CE_{M,\,Y}$	Emission Coefficient of Emission Y using Machine M	g/kg
$R_{M,\,j}$	Energy Consumption at Process j using Machine M	MJ
$E_{Y,j}$	Emission Component Y Generation at Process j	grams
TR_{System}	System Specific Energy Consumption	MJ
$TE_{Y,\,System}$	System Emission Component Y Generation	grams

The construction of the injection molding process component is similar to the industrial coating process component. The injection molding process has three different process types including a standard process, a standard process plus additional user-deified statistics and a complete user-defined process. Therefore, the logic flow behind the injection molding process component is the same as industrial coating process as shown in Figure 5-19. The pseudo codes to carry out the injection molding process component in listed in Algorithm 5-5.

Algorithm 5 - 5: Pseudo-code for the *Process with Emissions* Module
(Injection molding process component)

Step 1: Assign basic process information including:
action type, resources, and processing time;
Step 2: If the process is a standard process, go to Step 3;
If the process is a user-defined process, go to Step 5;
Step 3: Assign injection molding machine type M, and polymer shoot weight m_j;
Step 4: If there is additional user-defined statistics to collect, go to Step 5;
If no additional user-defined statistics, go to Step 6;
Step 5: Assign initial values for collecting user-defined statistics;
Step 6: Delay entity for an amount of processing time;
Step 7: If the process is a standard process, go to Step 8;
If the process is a user-defined process, go to Step 10;
Step 8: Set $f_M = m_j \cdot CR_M$, $g_Y = m_j \cdot CE_{M,Y}$;
$R_{M,\,j} = R_{M,\,j} + f_M$, $E_{Y,j} = E_{Y,j} + g_Y$;
$TR_{System} = TR_{System} + f_M$, $TE_{Y,\,System} = TE_{Y,\,System} + g_Y$;
Step 9: If there is additional user-defined statistics to collect, go to Step 10;
If no additional user-defined statistics, go to Step 11;
Step 10: Collect and store user-defined statistics;
Step 11: Continue simulation.

5.2.4 Process with Emissions Module Component for Plastics Processing

The injection molding process component in this research uses the study from using different molding machines. The plastic processing component created takes the approach from the material selection point of view. Ashby (2009) introduces material-based environmental impacts for materials including metals and alloys, polymers, and ceramics and glasses. CES EduPack is a piece of software which extends and supplements Ashby's book with more and detailed materials and their corresponding environmental metrics. Plastic processing, which is also included in CES EduPack, includes processes such as polymer production, extruding, size reduction, injection molding, and drying. The plastics processing component module is designed to examine the related environmental issues.

Figure 5-21 shows the interface of the plastics processing component of the *Process with Emissions* module. In the standard process option, the user will have to choose what type of plastic is processed. For thermoplastics, a drop down menu is shown with the options for thermoplastics. For the thermosetting plastics, a new drop down menu is shown to let the user choose which thermoset is used. Table 5-14 is a list of the plastics that are modeled in the plastics processing component module. Four types of plastics are modeled in this research but the list can be extended to hundreds of plastics using the data from CES EduPack 2010.

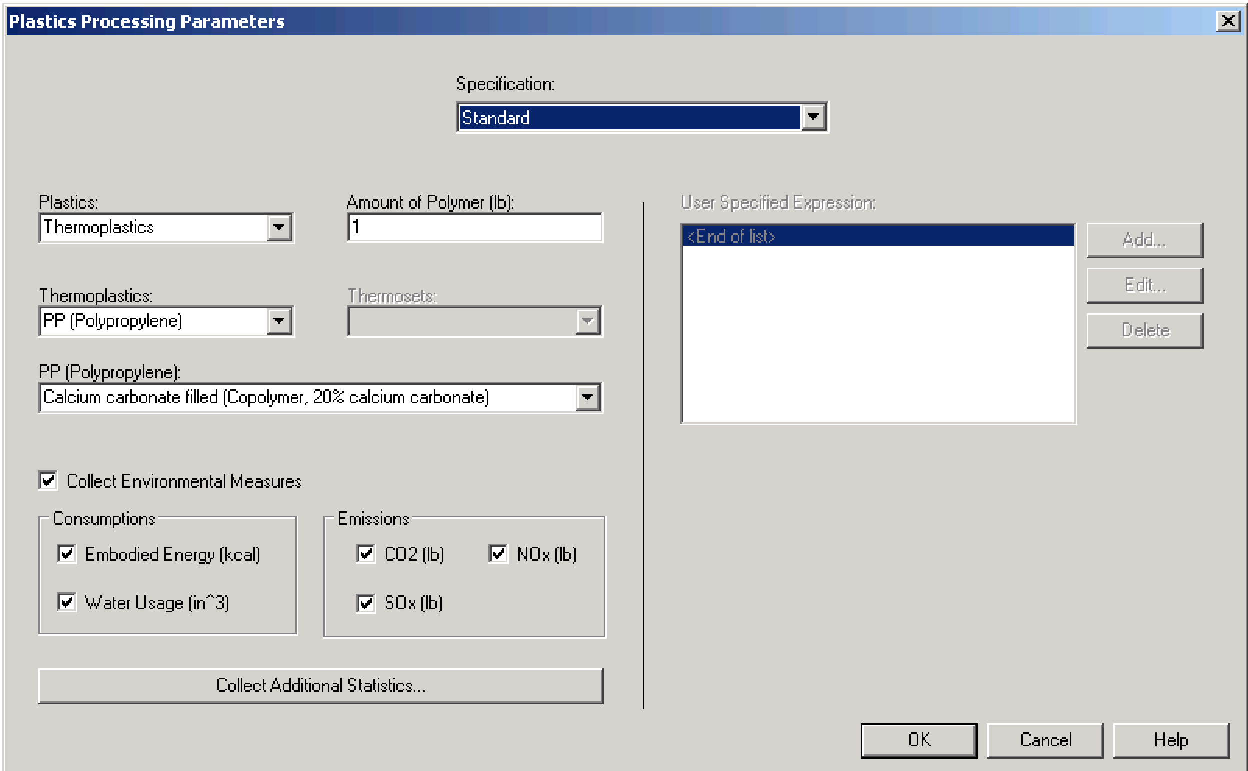


Figure 5 - 21: Plastics Processing Module Interface

Table 5 - 14: Types of Plastics include in the Plastics Processing

Thermoplastics	PP (Polypropylene)	• Calcium carbonate filled (Copolymer, 20% calcium carbonate) • Glass filled (30% long glass fiber)
	PVC (Polyvinyl Chloride)	• Glass filled (20% glass fiber, molding) • Unfilled (Chlorinated, molding and extrusion)
Thermosetting Polymers	PI (Polyimide)	• Glass filled (40% glass fiber) • Graphite filled (25% graphite)
	PUR (Polyurethane)	• Casting resin, unsaturated

The assumptions made for the plastics processing module component include:

1. The environmental performance measures collected are the average energy consumption and average emissions generation to process a particular type of plastic in general. It does not take into consideration of the processing details such as what type of machine is used for injection molding stage, or what type of machine is used for polymer extrusion stage.

2. It is assumed that the plastics processing does not take into consideration of the process efficiency.

For the plastics processing component, the environmental performance measures that the user is interested in knowing include the energy and water consumption, and the emissions generation. Simulation state variables are defined to represent the environmental performance measures. In particular, $R_{P,\,j}$ represents the energy consumed at process j to produce plastic P. $W_{P,\,j}$ is the state variable indicating the water usage at process j to produce plastic P. $E_{Y,j}$ stands for the amount of emission component Y generated at process j. TR_{System} is the system level state variable of energy consumption for the whole system. TW_{System} is the total water usage of the system and $TE_{Y,\ System}$ is the total emission type Y generation of the system. Table 5-15 summarizes the state variables and some system parameter notations used for the plastics processing process. CES EduPack has the results of the primary material production for its embodied energy consumption, water usage, carbon dioxide (CO_2) footprint, nitrogen oxides (NO_x) creation and sulfur oxide (SO_x) creation. All environmental performance measures, including the energy consumption (f), the water usage (h) and the emission generation (g), are a function of the amount of plastic processed. CES EduPack provides a list of energy coefficient (CR_P, in kcal per lb), water coefficient (CW_P, in in^3 per lb) and emission coefficient ($CE_{P,\,Y}$, in lb per lb) in order to perform the calculation. The mathematical formulation for calculating the energy and water consumption and emissions generation are specified as

$$f_P(m_j) = m_j \cdot CR_P$$

$$h_P(m_j) = m_j \cdot CW_P$$

$$g_{P,Y}(m_j) = m_j \cdot CE_{P,Y}$$

Table 5 - 15: Parameters and State Variables for Plastics Processing

Notation	Definition	Unit
P	Type of Plastic Processed	---
j	A specific Process	---
Y	Emission Component	---
m_j	Amount of Plastic Processed in Process j	lb
CR_P	Energy Coefficient for Plastic P	kcal/lb
CW_P	Water Coefficient for Plastic P	in^3/lb
$CE_{P,\,Y}$	Emission Coefficient of Emission Y for Plastic P	lb/lb
$R_{P,\,j}$	Energy Consumption at Process j to Process Plastic P	kcal
$W_{P,\,j}$	Water Usage at Process j to Process Plastic P	in^3
$E_{Y,j}$	Emission Component Y Generation at Process j	lb
TR_{System}	System Energy Consumption	kcal
TW_{System}	System Water Consumption	in^3
$TE_{Y,\,System}$	System Emission Component Y Generation	lb

The pseudo codes for executing the plastics processing component logic is listed in Algorithm 5-6.

Algorithm 5 - 6: Pseudo-code for the *Process with Emissions* Module
(Plastics processing component)

Step 1: Assign basic process information including:
action type, resources, and processing time;
Step 2: If the process is a standard process, go to Step 3;
If the process is a user-defined process, go to Step 5;
Step 3: Assign injection molding machine type M, and polymer shoot weight m_j;
Step 4: If there is additional user-defined statistics to collect, go to Step 5;
If no additional user-defined statistics, go to Step 6;
Step 5: Assign initial values for collecting user-defined statistics;
Step 6: Delay entity for an amount of processing time;
Step 7: If the process is a standard process, go to Step 8;
If the process is a user-defined process, go to Step 10;
Step 8: Set $f_P = m_j \cdot CR_P$, $h_P = m_j \cdot CW_P$, $g_{P,Y} = m_j \cdot CE_{P,Y}$;
$R_{P,\,j} = R_{P,\,j} + f_P$, $W_{P,\,j} = W_{P,\,j} + h_P$, $E_{Y,j} = E_{Y,j} + g_Y$;
$TR_{System} = TR_{System} + f_P$, $TW_{System} = TW_{System} + h_P$, $TE_{Y,\,System} = TE_{Y,\,System} + g_Y$;
Step 9: If there is additional user-defined statistics to collect, go to Step 10;
If no additional user-defined statistics, go to Step 11;
Step 10: Collect and store user-defined statistics;
Step 11: Continue simulation.

5.2.5 Capabilities and Limitations of the Process with Emissions Module

The *Process with Emissions* module is designed in order to capture the environmental performance measures for industrial and manufacturing processes. The module can model three industrial and manufacturing processes including the industrial coating process, the injection molding process and plastics processing process. In industrial coating process component, the module has the ability to collect and report the paint consumption, the solvent consumption and the VOC generation. The Injection molding component is able to determine how much energy is used and how much CO_2, SO_2, NO_X and CH_4 is emitted to the environment. The plastics processing component measures the amount of energy and water consumed in the process, and the amount of CO_2, SO_x, and NO_X generated as well. These environmental performance measures are reported to the user by each individual process and by the entire simulation system. The *Process with Emissions* module is designed to be easily used just as a standard process module to model processes in simulation. With some user input of process parameters, the user can model the industrial and manufacturing processes automatically. It is integrated part of the process module. Environmental performance measures are reported to the user at the end of the simulation as readily as traditional process performance measures such as resource utilization and average processing time. In addition to the standard environmental performance measures that are set up as the default measures, the users also have the ability to create and collect their own statistics under the *Process with Emissions* module developed.

Some of the limitations of the *Process with Emissions* module are listed as follows:

- The injection molding process component developed include the option of machine types only without specifying what kind of plastics is processed;
- The plastics processing component include only four types of plastics in this research;

- Assumptions are made that the emissions are generated constantly, however, the emission are collected discretely over time; and
- The *Process with Emissions* module models only three types of industrial and manufacturing processes.

5.2.6 Summary of the Process with Emissions Module

This section we have demonstrated the development of the sustainability toolkit for three different industrial and manufacturing processes. The *Process with Emissions* module is designed using the development methodology. Simulation models are generated for the test example for demonstration and toolkit verification and testing.

Some further work that can be done to the currently developed process with emissions module will be focused on the following areas:

- Obtain from other more recently studied literature to include options such as choosing one particular plastic to be processed in the injection molding process;
- Include more plastics options in the plastics processing component;
- Develop more industrial and manufacturing processes such as, but not limited to, commercial cooking, metals, chemical manufacturing, petroleum refineries, oil and gas production, pulp and paper, and cement manufacturing (EPA).

5.3 Material Handling Modules

The Environmental Protection Agency (EPA) estimates that forklift operations generated 297,973 tons of NOx, 64,892 tons of THC, and 1,357,677 tons of CO in 2007, which shows that the forklift operation is an important contributor to non-road emissions. Therefore, the material handling system is selected to be modeled. A set of material handling modules is designed

following the development methodology. To illustrate the development process of the materials handling modules, an illustration example (Example 5-3) is shown below.

Example 5-3: *Suppose a small company wants to evaluate the operations of their cross docking warehouse. A single forklift is used to transport the items among three stations which are the receiving station, intermediate station and the shipping station. The company is interested in knowing the environmental performance measures, such us emissions generation, in addition to traditional performance measures such as transporter utilization. The system layout animation is shown in Figure 5-22.*

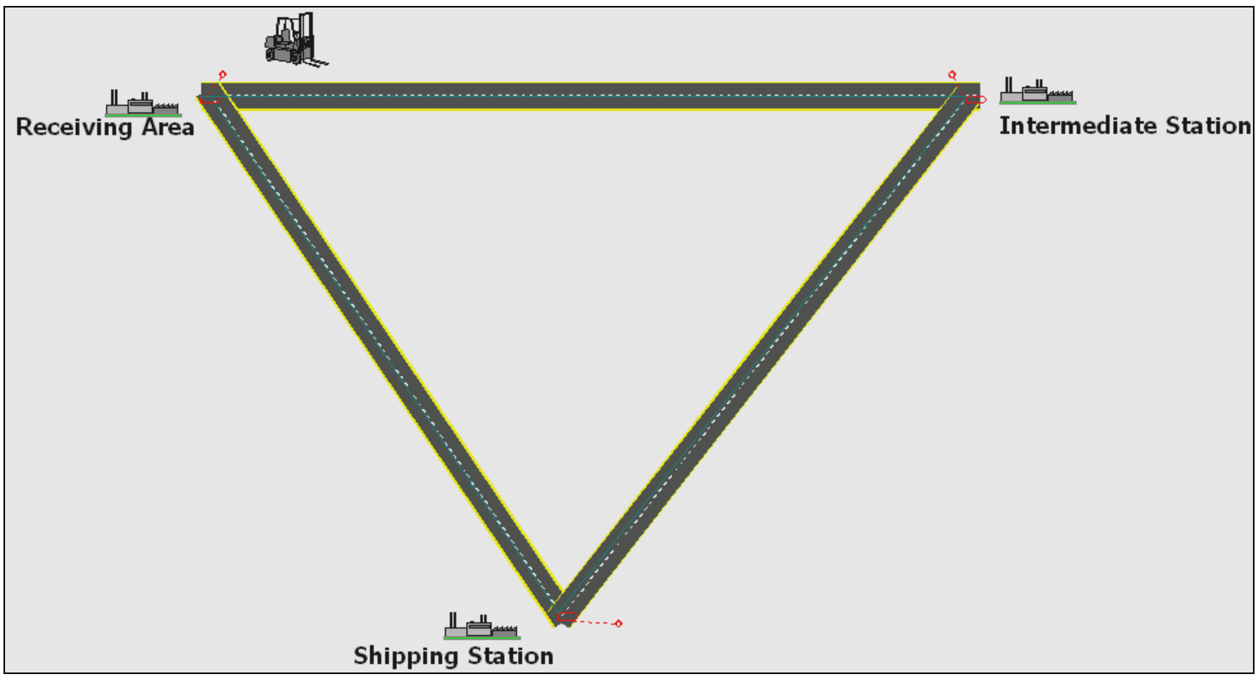


Figure 5 - 22: Animation of Sample Material Handling System

The material handling modules allow the modeler to model such a system to examine both traditional and suitability-based performance measures. The general development methodology is carried out step by step.

Identify the environmental performance measures and define the simulation state variable. Table 5-16 shows a summary of the parameter notations and the state variables used in the material handling modules.

Table 5 - 16: Parameters and State Variables for Material Handling Modules

Notation	Definition	Unit
h	Forklift Engine Horsepower	hp
d	Forklift Travelling Distance	feet
s	Forklift Average Speed	feet/hr
t	Interval Time when Forklift is On	minutes, hours
Y	Emission Component	---
C_Y	Emission Coefficient for Emission *Y*	grams/ hp·hr
τ	Forklift Load Factor	percentage
Idling	Forklift idling	
$E_{Y,Forklift}$	Emission Component *Y* Generation for a specified forklift	grams
$TE_{Y, System}$	System Emission Component *Y* Generation	grams

Two assumptions of the material handling system are made including the following:

1. The transporter is travelling at the average speed; and
2. Emissions generated per unit of time are the same regardless of the transporter status (travelling or idling).

Since the forklifts burn fuel while it is running, the environmental performance measures mainly focus on the air pollution generation. The state variable representing the emissions generation for emission component *Y* for a single forklift is defined as $E_{Y, Forklift}$. A system level state variable representing the sum of total emissions generation among all forklifts used in the system is defined as $TE_{System.}$

Determine the simulation events. In the simulation model, the forklift is controlled by the simulation entity (products, inventory, ect.). Forklifts will be moving around in two circumstances. The forklift is requested by the entity and is running from its current location in the system to the location where the entity sends the request. And the forklift is carrying the entity to the entity's destination location. The simulation system consists of four types of events

and they are the requesting transporter event, the transporting event, the arrival at destination event and the freeing transporter event.

When an entity needs to be transported, it sends out a request and allocates the transporter (forklift in this case). After the forklift is allocated, the requesting transporter event is triggered and the transporter starts to move to the entity's request location. The request event triggers the movement of the transporter therefore triggers the calculation of environmental performance measures. After the transporter arrives at the entity's request location, it is determined to be left idling or turned off. If the transporter is left idling, the request event also triggers the start collection of emissions.

The transporting event travels the forklift to the entity's destination location along with the entity. This event triggers the end collection of environmental statistics if the transporter is left idling from the request event. The transport event also triggers the start collection of the environmental statistics during its travelling time.

The arrival at destination event means that the transporter arrives at the entity's destination location along with the entity. The location is represented as a station in simulation modeling. Once the arrival event happens, it will end the collection of emission statistics that are being updated along the route where the transporter travels from. It determines whether or not the transporter is left idling and it triggers the collection again if the transporter is left idling.

The freeing transporter event is carried out when the entity frees the associated transporter. If the transporter happens to be left idling when it is freed, the free event will calculate and update the environmental statistics for the period of time that the transporter is idling.

Formulate the mathematical relationship. U.S. Environmental Protection Agency (EPA, 2004) provides a formulation to calculate the emissions generated from nonroad engines. In the material handling system, this formulation is translated to the following equation

$$f_Y(t) = C_Y h t \tau$$

where the generation of emission component Y (f_Y) equals the production of engine power h, engine running time t, emission coefficient C_Y and the forklift load factor τ. When the forklift is traveling, the time factor can be replaced by distance d over average speed s resulting the following equation:

$$f_Y(d) = \frac{C_Y h d \tau}{s}.$$

EPA report (2010) and EPA report (2002) provide a list of the emission coefficient C_Y for spark-ignition engines (>25 hp) which is commonly used in industrial equipment. The list contains the emission coefficient for air pollution including hydrocarbon (HC), carbon monoxide (CO), nitrogen oxide (NO_x) and particulate matter (PM). EPA report (2004) provides a list of the load factor τ for forklift engines as well. This research uses these two lists provided by EPA to determine the emissions generation for these four types of emissions.

To calculate the emission generation for a particular forklift, two parts have been summed, one of which is the total emission from forklift running on road, and another one is total emission when forklift is left idling. We use $d_{i,Forklift}$ to represent all the travelling distance components of the forklift, and $t_{j,Forklift}$ to represent the idling time components of the forklift. Emission component Y generation for a specified forklift $E_{Y,Forklift}$ can be calculated as

$$E_{Y,Forklift} = \sum_i f_Y(d_{i,Forklift}) + \sum_j f_Y(t_{j,Forklift}).$$

The system level performance measures sums up the forklift based performance measures for all forklift used within the system therefore can be formulated as

$$TE_{Y\,,\,System} = \sum_{Forklift} E_{Y\,,Forklift} \,.$$

5.3.1 Material Handling Modules Development

Based on the events identified and mathematical relationship specified, four flow chart modules and one data module have been created for the material handling modules. These modules include the *Request with Emissions* module, the *Transport with Emissions* module, the *Free with Emissions* module, the *Transporter Station* module, and the *Transporter Information* data module. Figure 5-23 displays the material handling modules developed from the sustainability toolkit.

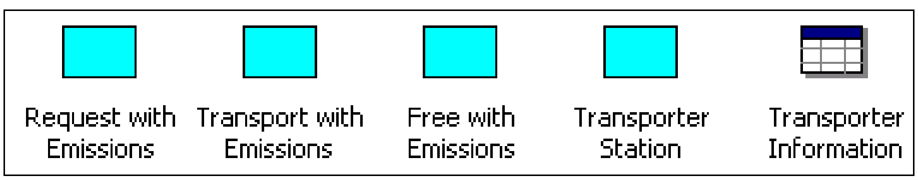


Figure 5 - 23: Material Handling Modules

The *Request with Emissions* module (see Figure 5-24(a)) has the full functionality of the standard Request module. In addition to the parameters to be entered by the user, such as the transporter name, selection rule, and velocity, the module asks the user to provide an idling probability that determines the status of the forklift when it arrives at the entity's location. The module allocates the forklift to be used and calculates the emissions generated along the route from its original location to the location where the entity requests it.

The *Transport with Emissions* module (see Figure 5-24(b)) has the full functionality of the standard Transport module. The user is asked to specify the station to which the entity is transported. The user also needs to specify the load factor for that movement from the current

location to the destination location, which will overwrite the one specified in the transporter information data module. The *Transport with Emissions* module can calculate the emissions generated during the period of time after the forklift arrives at the location if the forklift is left idling.

The *Transporter Station* module (see Figure 5-24(c)) provides the same functionality (representing a physical location) as a Station module. Additionally, when the forklift arrives at the entity's destination, which is modeled as a Station, our *Transporter Station* module collects the emissions generated along the route from the location of the previous location where the entity is transported from. Finally, the user can specify the probability of the truck idling while it remains at the station location.

The *Free with Emissions* module (see Figure 5-24(d)) frees the Forklift that is used to transport the entity. It collects the emissions generated if the forklift is left idling while it remains at the station.

The *Transporter Information* data module (see Figure 5-24(e)) asks the user to specify the necessary parameters needed in order to make the calculation for environmental performance measures. For a forklift, it asks the user to enter the rated horsepower and the load factor.

To demonstrate the development process, a specified material handling system configuration is defined. A single forklift is used in the system. The engine G2GT25 is chosen to be used for the forklift from EPA report (2010) and EPA report (2002). The engine power is specified as 288 horsepower. The load factor of the forklift obtained from EPA (2004) is specified as 0.3.

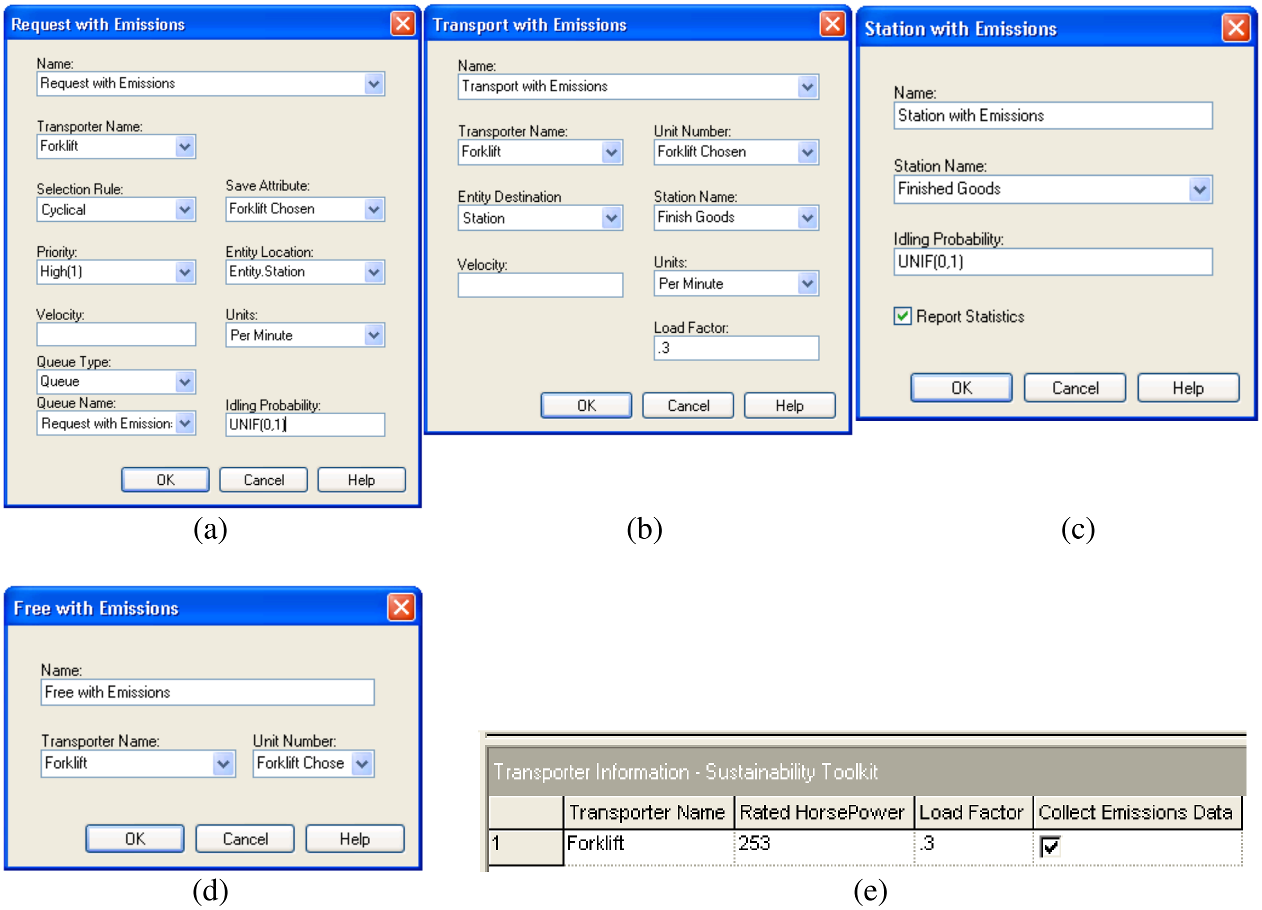


Figure 5 - 24: Material Handling Modules Dialogue Boxes: (a) *Request with Emissions* Module, (b) *Transport with Emissions* Module, (c) *Transporter Station* Module, (d) *Free with Emissions* Module, and (e) *Transporter Information* Date Module.

With system specifications determined, a simulation model is constructed using the standard modules (see Figure 5-25). The logic is straight forward. The entity is created and enters the receiving station. It then requires the forklift to come to the receiving station to pick the entity up. Environmental performance measures start to be collected when the forklift is moving toward the entity. Then the entity is uploaded to the forklift, during which period of time, the forklift might be left idling and keep on generating emissions. The entity is then transferred to the intermediate station using a transport module. After it arrives at the intermediate station, environmental statistics are first updated. The entity is then dropped off at the intermediate location with forklift left idling or turned off. When the entity gets off from the forklift, it frees

the forklift by a free module. Statistics are kept updating if the forklift is left idling during the dropping-off process. Then the entity requests the forklift once again and the logic flow follows the same pattern.

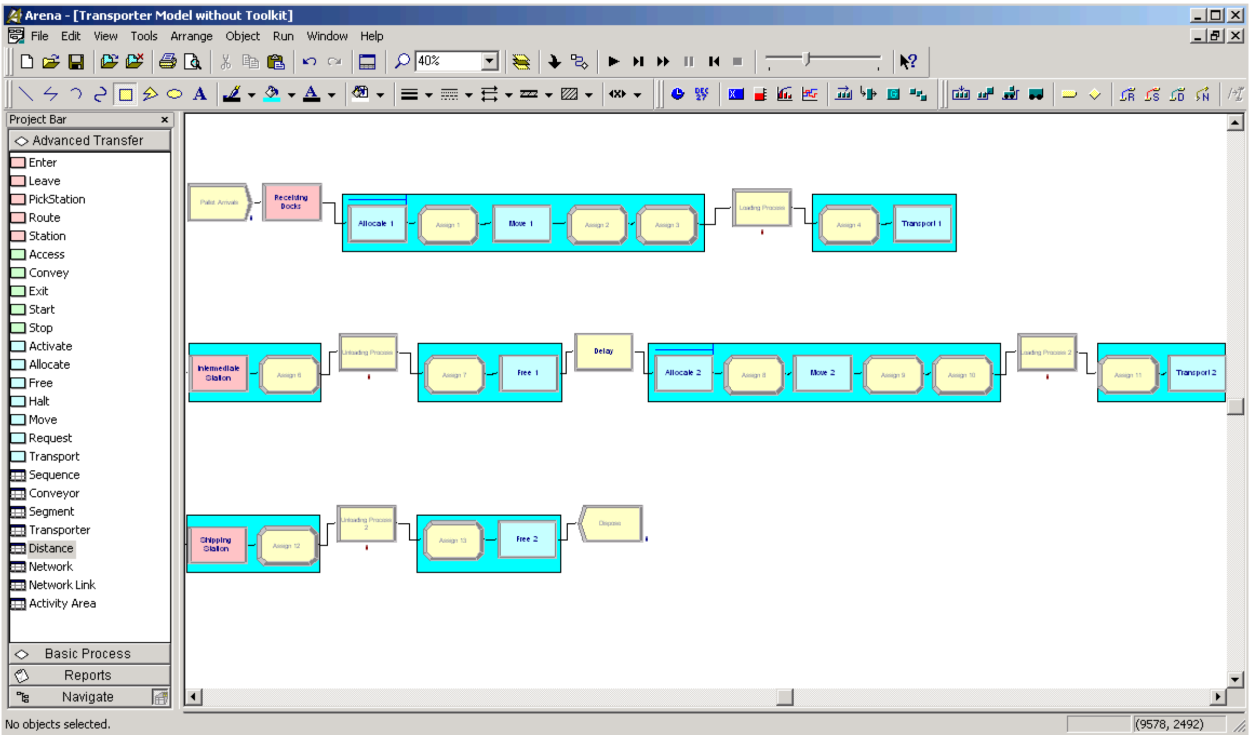


Figure 5 - 25: Simulation Overview for Material Handling System using Standard Modules

The environmental performance measures are calculated and updated within the green boxes in Figure 5-25 of the original model. The mathematical relationship and the logical relationship in those boxes are able to be reconstructed and integrated into the material handling modules. Next part of the paper we introduce each one of the material handling modules for how they are created.

5.3.1.1 Transporter Information Data Module

The transporter information data module is used to assign transporter information including transporter name, rated horsepower and load factor (see Figure 5-26). The corresponding system variables of forklift power and load factor have their initial value assigned.

Checkbox of collecting emission data is checked if the user wants to return default environmental performance measures.

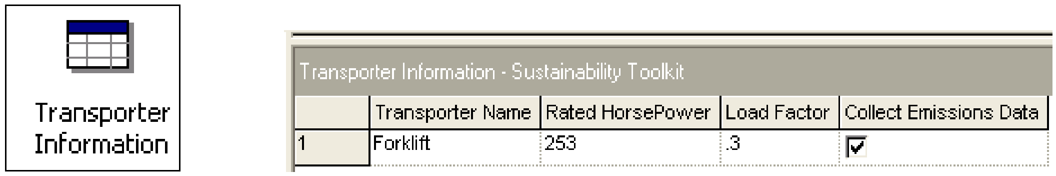


Figure 5 - 26: Material Handling Modules: *Transporter Information* Data Module

5.3.1.2 Request with Emissions Module

The *Request with Emissions* module is used when an entity requests a forklift to cover over and pick-up the entity (see Figure 5-27). The module provides the same functionality as a standard Request module where the user can specify the transporter name, selection rule if there are multiple transporters, save attribute which saves the selected transporter number to an attribute, and velocity. In addition to the standard Request module, the *Request with Emissions* has a user-input field for entering the transporter idling probability after the transporter reaches the requesting entity. The module also captures and updates the amount of emissions generated from running the transporter to come over. The logic flow in the *Request with Emissions* module is described in Figure 5-28. The entity first allocates the transporter. If the transporter is currently busy, the entity will wait in a queue. Once the transporter is allocated, it is moved towards the requesting entity. Collection of environmental measures is triggered to start. When the transporter reaches the requesting entity, the emissions created along the way of the transporter is collected and updated. The entity may carry out other simulation operations such as loading process while the transporter is left idling or turned off. Therefore, at the end of the *Request with Emissions* module, trucking idling parameter is assigned.

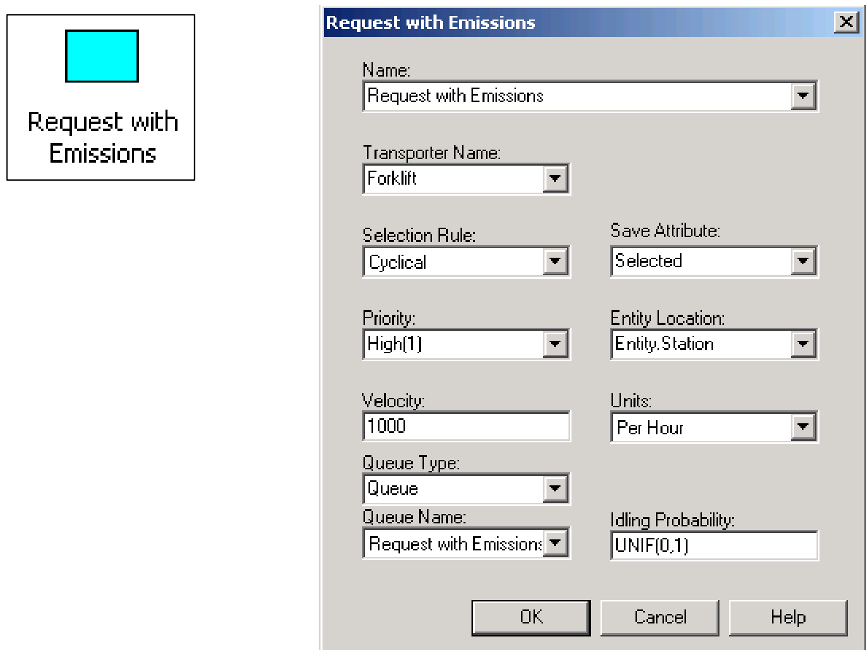


Figure 5 - 27: Material Handling Modules: *Request with Emissions* Module

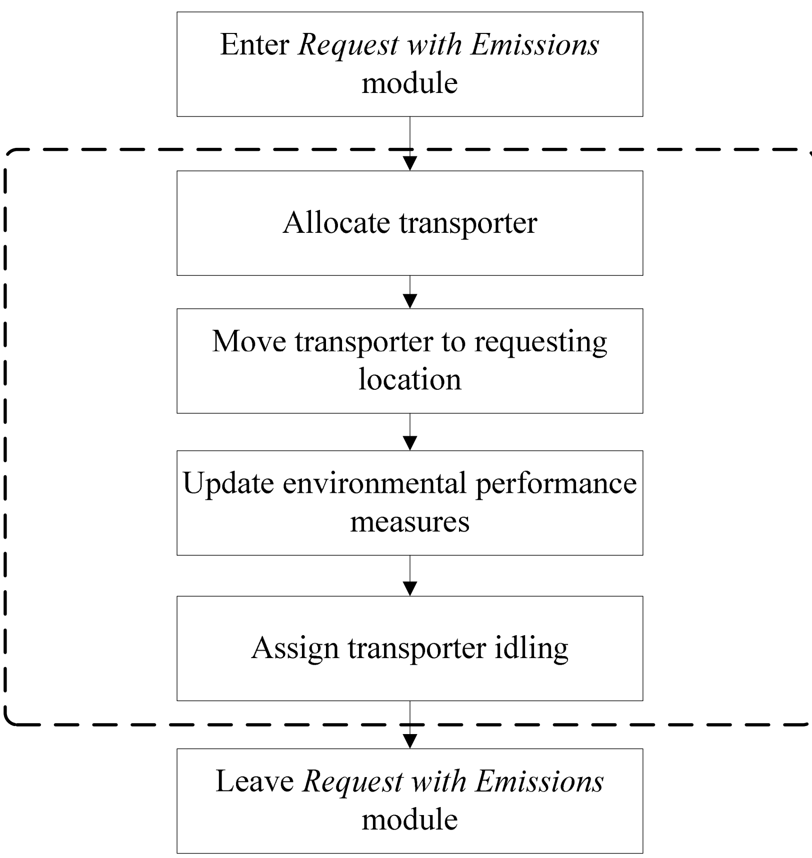


Figure 5 - 28: Flowchart of the logic for the *Request with Emission* Module

The pseudo codes for executing the *Request with Emissions* module is listed in Algorithm 5-7.

Algorithm 5 - 7: Pseudo-code executed in the *Request with Emission* Module

Step 1: Entity allocates the transporter;
Step 2: Set t_{move_start} = TNOW*;
Entity requests and moves the transporter to come over;
Step 3: Set t_{move_end} = TNOW;
Set $f_Y(t) = C_Y h\tau(t_{move\_end} - t_{move\_start})$;
Set $E_{Y,Forklift} = E_{Y,Forklift} + f_Y$;
Set $TE_{Y,Forklift} = TE_{Y,Forklift} + f_Y$;
Set t_{upload_start} = TNOW;
Step 5: Set *Idling* = 0 or 1;
Step 6: Continue simulation.

* TNOW means current simulation time

5.3.1.3 Transport with Emissions Module

The *Transport with Emissions* module is used to transport the entity to its destination location (see Figure 5-29). The module functions the same as a standard Transport module where the user specifies on what transporter is the entity transported. And it specifies what the entity's destination station is. Additionally, the module can specify the load factor of the transporter which overwrites the value obtained from the *Transporter Information* data module.

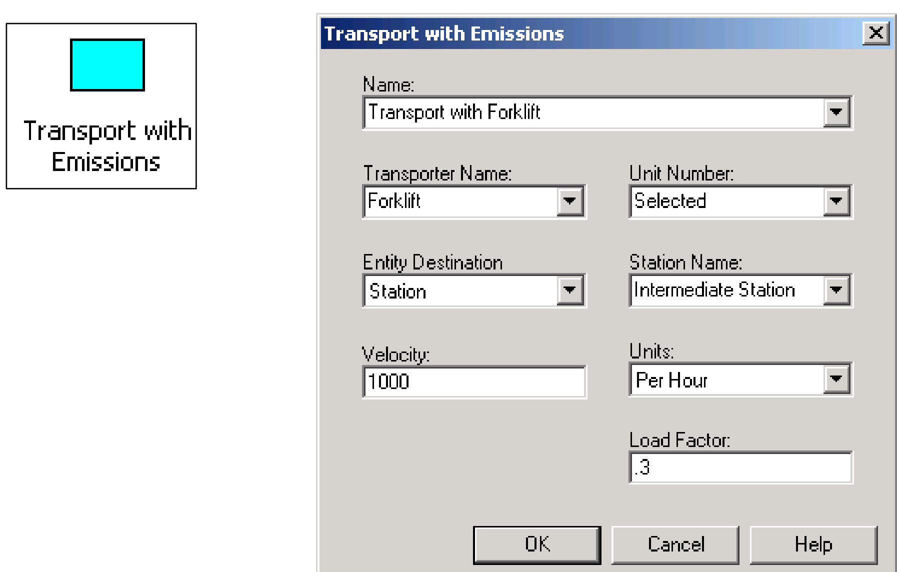


Figure 5 - 29: Material Handling Modules: *Transport with Emissions* Module

During the period of time from the transporter's arrival at the requesting entity, to the entity's transporting by the transporter using the *Transport with Emissions* module, if the transporter is left idling, the emissions are being generated along the time as well. Depending on the transporter attribute 'idling', the *Transport with Emissions* module collects and updates the emissions generation. A typical process happening during this time of period will be a loading process which loads the entity onto the transporter. Therefore, we use a upload start and end time to capture the time interval for calculation. The logic flow is shown in Figure 5-30 and the pseudo codes representing the logic flow are shown in Algorithm 5-8.

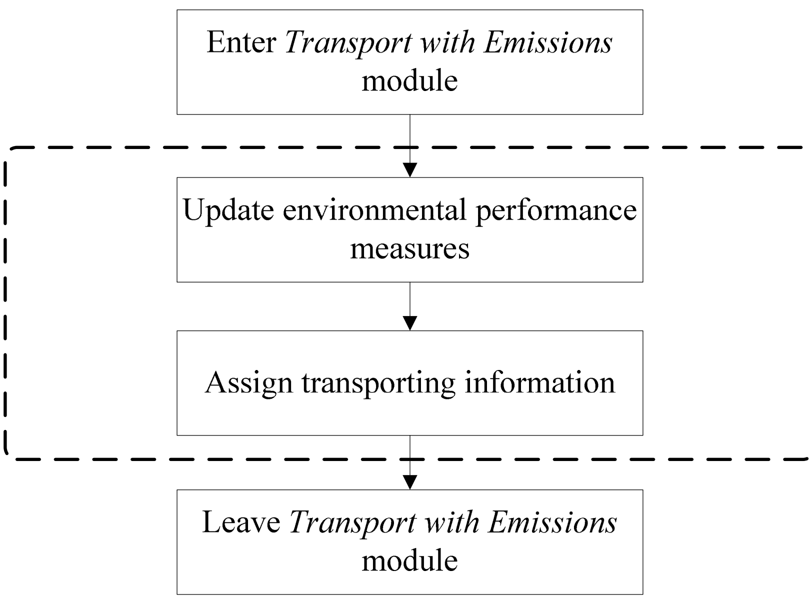


Figure 5 - 30: Flowchart of the logic for the *Transport with Emission* Module

Algorithm 5 - 8: Pseudo-code executed in the *Transport with Emission* Module

Step 1: Set t_{upload_end} = TNOW;
Set $f_Y(t) = Idling \times C_Y h\tau(t_{load\_end} - t_{load\_start})$;
Set $E_{Y,Forklift} = E_{Y,Forklift} + f_Y$;
Set $TE_{Y,Forklift} = TE_{Y,Forklift} + f_Y$;
Set $t_{transport\_start}$ = TNOW;
Step 2: Assign route information such as the destination station name and the transporter velocity;
Step 3: Continue simulation.

5.3.1.4 Transporter Station Module

The *Transporter Station* module is used to model a physical location in simulation (see Figure 5-31). When transporter arrives at the entity's destination location, modeled as a transporter station module, the module collects the emissions generated along the route from the previous location. The module also determines whether the transporter is left idling after it remains at the station.

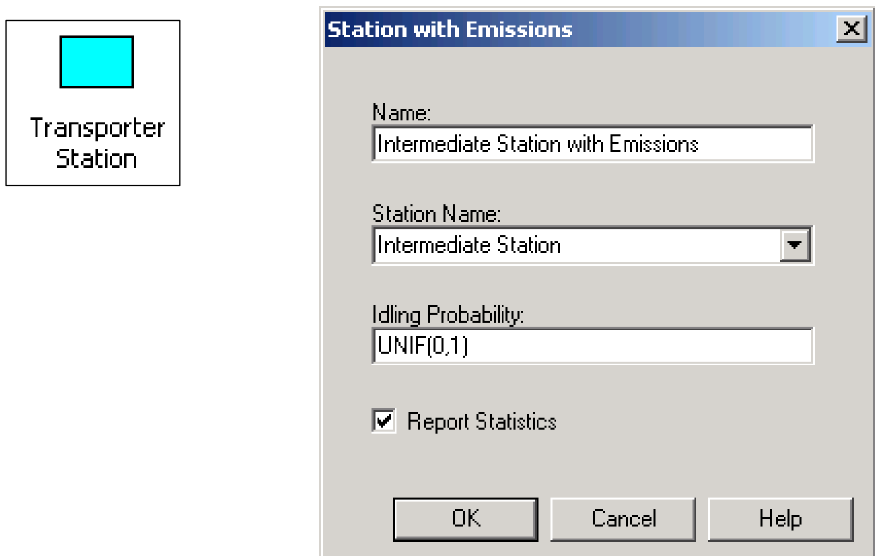


Figure 5 - 31: Material Handling Modules: *Transport Station* Module

Figure 5-32 depicts the logic flow of the *Transporter Station* module. When the entity arrives at the station, it determines how long has the transporter been travelled. The time interval is used to calculation the emissions generation. Both transporter-based and system-based environmental variables are updated using the equations specified in previous section. The pseudo codes are shown in Algorithm 5-9.

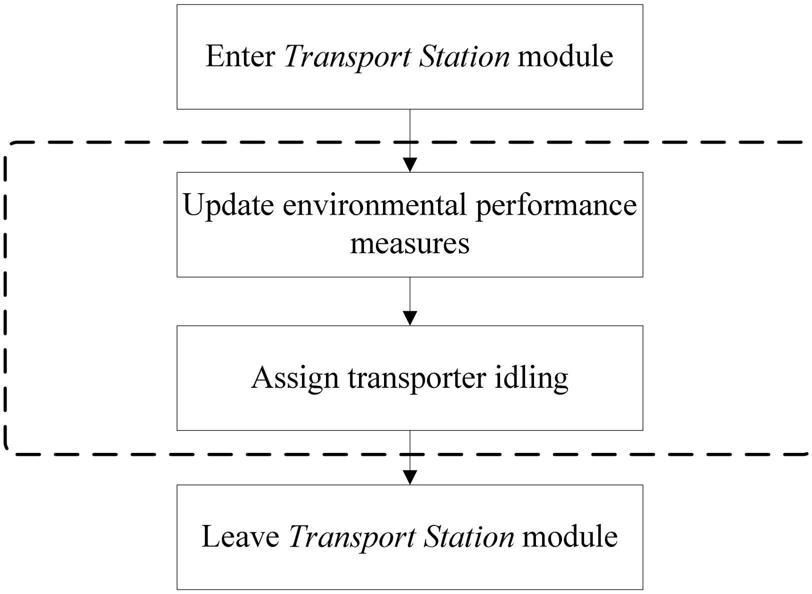


Figure 5 - 32: Flowchart of the logic for the *Transporter Station* Module

Algorithm 5 - 9: Pseudo-code executed in the *Transporter Station* Module

Step 1: Set $t_{transport\_end}$ = TNOW;
Step 2: Set $f_Y(t) = C_Y h\tau(t_{transport\_end} - t_{transport\_start})$;
Set $E_{Y,Forklift} = E_{Y,Forklift} + f_Y$;
Set $TE_{Y,Forklift} = TE_{Y,Forklift} + f_Y$;
Set t_{unload_start} = TNOW;
Step 3: Set *Idling* = 0 or 1;
Step 4: Continue simulation.

5.3.1.5 Free with Emissions Module

The *Free with Emissions* module is used for the entity to free the transporter once it is delivered to its destination location (see Figure 5-33). The module has the same interface as a standard Free module. User only needs to specify which transporter to free. When the transporter is left idling after it arrives at the entity's destination location, emissions are continuously emitted. The *Free with Emissions* is able to collect the emissions creation under such circumstances.

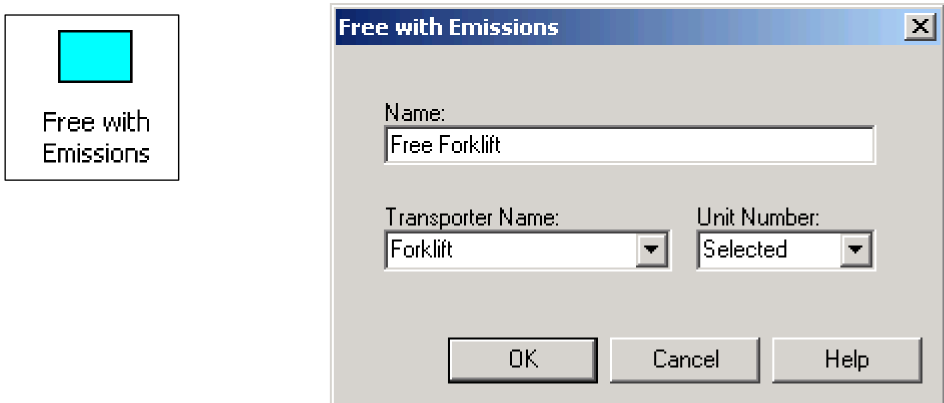


Figure 5 - 33: Material Handling Modules: *Free with Emissions* Module

Figure 5-34 shows the logic flow and Algorithm 5-10 is the pseudo codes for the *Free with Emission* module.

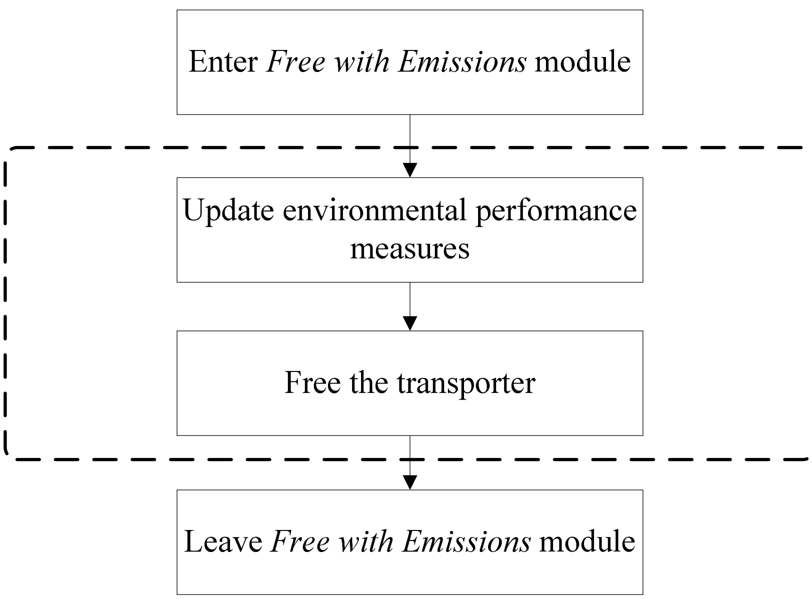


Figure 5 - 34: Flowchart of the logic for the *Free with Emission* Module

Algorithm 5 - 10: Pseudo-code executed in the *Free with Emission* Module

Step 1: Set t_{unload_end} = TNOW;

Set $f_Y(t) = Idling \times C_Y h\tau(t_{unload\_end} - t_{unload\_start})$;

Set $E_{Y,Forklift} = E_{Y,Forklift} + f_Y$;

Set $TE_{Y,Forklift} = TE_{Y,Forklift} + f_Y$;

Step 2: Free the transporter;

Step 3: Continue simulation.

5.3.1.6 Model using the Material Handling Modules

The simulation model constructed to represent the material handling system using the standard modules is remodeled using the material handling modules developed. All the system configurations and specifications remain the same. Figure 5-35 displays the overall simulation model using the material handling modules.

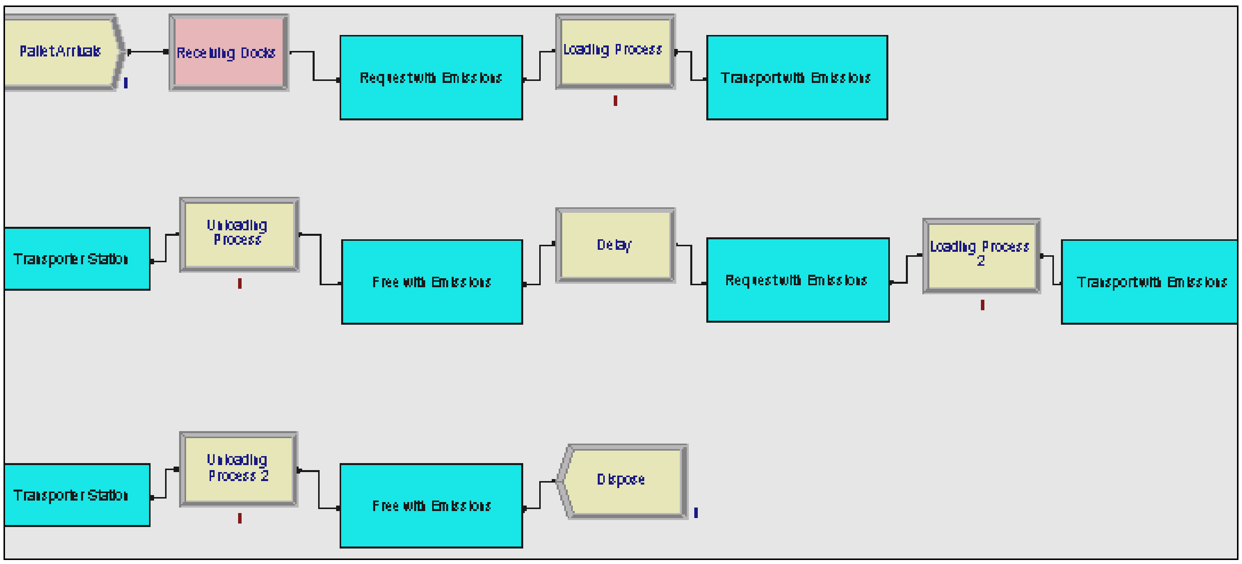


Figure 5-35: Example 5-3 Simulation Model using *Material Handling* modules

The boxed areas from the original model have now been replaced by the material handling modules. To illustrate the uses of the material handling modules to collect emissions generation, the model constructed using the material handling modules is run for 220 hours simulation time length and the results from running the model are shown in Table 5-17. This is a typical example output that the user will get using the material handling modules.

Table 5-17: Simulation Results from running Transporter Model

Environmental Performance Measures	Value
HC Emissions (gram)	1.95 E+05
CO Emissions (gram)	4.57 E+05
NO_x Emissions (gram)	272.73
PM Emissions (gram)	7241.60

5.3.2 Verification, Validation and Testing of the Material Handling Modules

Verification of the *Material Handling* modules is to examine that whether the development methodology has been translated to a proper computer toolkit. After debugging the *Material Handling* modules along its development process, no error or warning has been found using these modules to perform simulation-based modeling. The modules work as desired. The warehouse system has been modeled and the environmental performance measures, specified as the emissions generation, are collected and reported to the user. Testing of the module contains the comparison for the original simulation model using standard modules and the simulation model using the material handling modules. The main function of the material handling modules is to determine the environmental performance measures as easily and efficiently as other built-in traditional measures. Therefore, the testing of the modules is to prove that there is no statistical difference on these two simulation models. Table 5-18 shows a module testing of 95% confidence interval on the emissions generation for the original model without using the toolkit and for the simulation model using the toolkit. Both models are run for 100 replicates in order to construct the confidence interval. As a result, the material handling modules provides reasonably the same results. A two-sample-t test is performed to compare the difference of the mean value of emissions generated from both models. Using Arena Output Analyzer, Table 5-18 also shows that no statistical differences have been found among these two models. Therefore, the material handling modules are tested and verified to work as desired.

Table 5 - 18: Module Testing of 95% Confidence Intervals on Emissions Generation

	Original Model without Toolkit	Model with Toolkit	Value Difference	
	Average	Average	95% CI Min Value	95% CI Max Value
HC Emissions	2.13E+05	2.11E+05	-6730	1950
CO Emissions	4.98E+05	4.93E+05	-1.57E+04	4.56E+03
NO_x Emissions	297.26	293.93	-9.39	2.72
PM Emissions	7892.80	7804.30	-249	72.2

5.3.3 Capabilities and Limitations of the Material Handling Modules

In this section, the *Material Handling* modules have been created to model warehouse material handling systems. The modules are able to model the main transporter operations including the request of a transporter, the transporting action, and the freeing transporter action. A study of the industry used forklift is used and the material handling modules are able to examine and report the emissions of hydrocarbon (HC), carbon monoxide (CO), nitrogen oxide (NO_x) and particulate matter (PM) generated by using the forklift. The modules are implemented as an integrated part where the user only needs to specify the transporter information in a data module and use the flow chart modules just as the standard modules. Simulation report of the emissions generated is displayed automatically to the user at the end of simulation.

Some of the limitations associated with the *Material Handling* modules are listed as follows:

- The material handling modules only model one single forklift;
- The default environmental performance measures only include four types of air pollution;
- The material handling modules do not have the flexibility to collect user-defined environmental statistics; and

- Assumptions are made that the emissions are generated constantly, however, the emission are collected discretely over time.

5.3.4 Summary of the Material Handling Modules

The toolkit development methodology is applied to warehouse material handling systems to create material handling modules. A prototype example is modeled using the developed modules to for demonstration and for testing. The material handling modules are able to efficiently measure, in this case, the emissions generated from the handling system operations.

The future work for continuously developing the material handling modules includes the following aspects:

- Add the capability to the modules that multiple forklifts can be used;
- Add the capability to the modules that different types of transporters can be used;
- Obtain from other research to provide more options in collecting environmental performance measures (energy consumption, electric usage, etc.); and
- Add user-specified statistics collection mechanism to the current material handling modules.

5.4 General Collection Modules

The sustainability toolkit modules introduced in the previous sections model particular systems of interest. 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 emissions generated in warehouse operations where transporters such as forklifts are used. *Process with Emissions* module is used to collect the environmental measures from industrial and manufacturing processes. These modules are

designed in such a way that each module models one type of system. The question is then asked, "Can a general collection toolkit be designed, so that it can be used in any systems of interest and collect user-defined environmental measures through the simulation model?" As a result, this research develops a generalized sustainability toolkit.

In order to generalize the environmental performance measures collection mechanism, the following two steps need to be performed:

- Generalization of the event triggers that starts and ends the collection;
- Generalization of the user-specified data inputting.

To demonstrate the process of developing general toolkit modules, a simplified example of Example 5-1 is specified.

Example 5-4: *Suppose the same company which has a centralized warehouse facility and makes deliveries to customers on a daily basis. The company currently has delivery trucks that consume diesel fuel. Each day routes are assigned to trucks. The route may consist of a variable number of delivery stops at customers and variable route distances that can each be represented by a probability distribution. The company is interested in using their own calculation formula to evaluate the carbon dioxide (CO_2) and nitrogen oxides (NO_x) generated by each individual truck and by the entire system.*

To generalize the collection mechanism – every collection has a start and an end. Figure 5-36 summarizes the general steps of collecting environmental performance measures, which is represented by emission generation in this case. First, the begin event trigger initializes the emission event to start. Emission is generated over some elapsed time and space. At the end, there needs to be another end event trigger that ends the emission event to stop collecting emission generation. The statistics is performed at the last end event trigger simultaneously when

the trigger stops the emission event. Using the modified transportation system as an example, when truck leaves the warehouse and runs toward the customer, it is when the event trigger kicks in and starts the emission generation collection. Emission is continuously generated when truck is running. Truck arrives at the customer location and is turned off. Emission no longer is generated so the end event trigger kicks in and stops the emission generation. Statistics is calculation and stored for future usage.

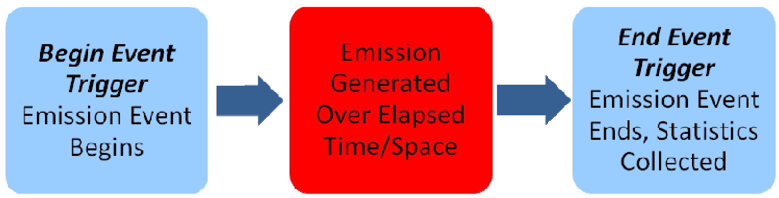


Figure 5 - 36: Three Steps of Collecting Emission Generation

Figure 5-36 carries out the generalization of the event triggers that starts and ends the collection. As a result, a *Begin Collection* module is developed to start the collection of environmental performance measures while an *End Collection* module is developed to end the collection in the simulation model. They are shown in Figure 5-37.

A data specification module also needs to be developed to provide the user with a consistent and regulated way to enter the calculation formula and data. A data module *Collection* is developed and showed in Figure 5-37 to supplement the design of general collection toolkit. Details of these general modules are discussed in Section 5.4.1.

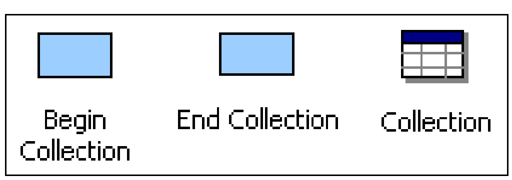


Figure 5 - 37: General collection Modules

5.4.1 General Collection Modules Development

In this section the details of developing the general collection modules and how to use them are discussed. The three collecting steps previously presented in Figure 5-36 not only apply for the emission generation, but also apply for other environmental performance measures. The generalized collection mechanism is the same. A performance measure (collection) is defined in *Collection* data module. *Begin Collection* module is used in the simulation model to trigger the beginning of the collection, and *End Collection* module is used to trigger the ending of the collection.

5.4.1.1 Collection Data Module

The *Collection* data module is used to specify the sustainability performance measure (collection). Figure 5-38 depicts the *Collection* data module created. The user can define a performance measure and the formula of how to calculate and update such a measurement.

Collection - Sustainability Toolkit

	Collection Name	Collection Type	Rows	Member Value	Delta	Delta Type	Accumulator Expression
1	CO2 generation	Entity Accumulator	8	8 rows	☑	TNOW	Delta * Truck HP * Truck Ideling * EmissionCoe(1)
2	NOx generation	Entity Accumulator	8	8 rows	☑	TNOW	Delta * Truck HP * Truck Ideling * EmissionCoe(2)
3	System CO2 generation	System Accumulator		0 rows	☐	TNOW	CO2 generated by truck
4	System NOx generation	System Accumulator		0 rows	☐	TNOW	NOx generated by truck

Figure 5 - 38: General Collection Modules: *Collection* data module

In the first field, a collection name is given to the performance measure defined. The *Collection* data module allows for two types of performance measures which can be specified in the collection type tab. The two types of performance measures are entity specific (Entity Accumulator) or system performance measures (System Accumulator). These two types of measures are introduced separately.

System Accumulator

A system accumulator is a performance measure that collects the defined measurement

for the entire simulation system. In Example 5.4, the system accumulators are system CO_2 generation and system NO_x generation. The “Rows” and “Member Value” tabs are used to specify the entities that trigger the collection events. Since system accumulator applies for all the entities that trigger the collection events, these two tabs are not applicable with the system accumulator and will be introduced in the entity accumulator. The “Delta” and “Delta Type” are used to define the “Accumulator Expression” tab as needed and will be introduced in detail. The “Accumulator Expression” tab specifies how to evaluate the incremental value that adds up to the system accumulator when the end collection event is carried out in the model. The accumulator expression can be an attribute, a variable or a mathematical expression. Once a system accumulation is created by the user, a unique system variable is created to present the defined system accumulator. When corresponding end collection event is executed in the model, the accumulator expression is evaluated and the incremental value is calculated and stored in the system accumulator variable. At the end of the simulation, the final value of the system accumulator variable is returned to the user.

Entity Accumulator

An entity accumulator is used to collect environmental performance measures for multiple entities, each of which is collected individually. Using Example 5-4 as an example, the system CO_2 and NOx emissions are selected using the system accumulators. The company is also interested in knowing the CO_2 and NOx emissions generated by each of the eight trucks. Entity accumulators can be used for this purpose. Two separate entity accumulators, namely the CO_2 generation and the NO_x generation, are created to collect the emissions individually by truck (see Figure 5-38). The “Rows” and “Member Value” tabs are used to define the entity collection array where “Rows” tab indexes the collecting entities and “Member Value” sets up the

matching values. The entity accumulator will be activated if and only if the following two conditions are met:

- The entity has one of the matching values; in other words, the entity falls into the collection array; and
- The entity carries out the corresponding end collection event.

The accumulator expression is set up similarly as in the system accumulator. The same expression is used for all entity members specified in the array. Once an entity accumulator is created, a unique simulation variable array is created. When the above two conditions are met and an entity (index *i*) triggers the end collection event, the accumulator expression is evaluated and the incremental value is calculated and added to the variable array (index *i*). At the end of the simulation, the final value of the entity accumulator variable array is returned to the user.

The entity accumulator is also powerful to collect a group of entities as one collection index. For instance, in Example 5-4, if trucks 1, 2, 3 and 4 consume diesel and trucks 5, 6, 7, and 8 use 20% soya blend biodiesel (B20), the company is interested in seeing the emissions generated by using these two types of fuel. Entity accumulators can be created having two array entries (index 1 and index 2) and the member value is set to be the fuel type (index 1 as for diesel and index 2 as for B20). Entities of trucks 1, 2, 3, and 4 will be in index 1 array entry while entities of trucks 5, 6, 7, and 8 will be in index 2 array entry.

Accumulator Expression

The “Delta” and “Delta Type” tabs are used to be included in the accumulator expression and provide the user with formula inputting flexibility. It represents the difference of a specified expression. Two commonly used delta types are introduced as follows:

- Difference in time: From the collection starting event to the collection ending event, the elapsed time is used as an inputting parameter.
- Difference in values: This type of delta includes other value difference from the collection starting event to the collection ending event. To demonstrate this type of delta, it is assumed that the simulation entities are trays of unfinished products. The unfinished products go through a painting process and the emissions measurement is a function of the total number of products painted in the tray and some other parameters. The demonstration is shown in Figure 5-39.

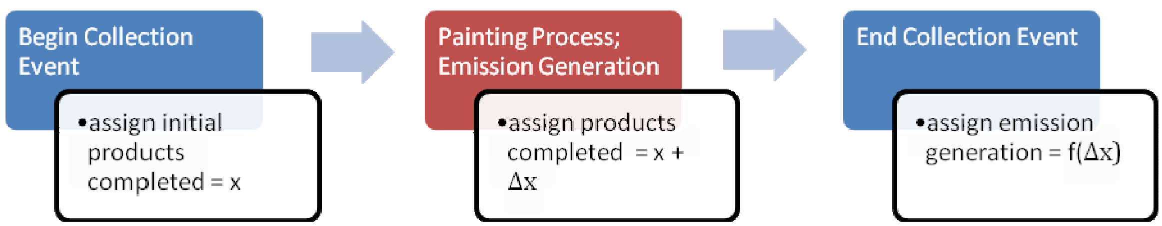


Figure 5 - 39: Delta Type of Difference in Values

As a result, three options of the delta types are created and listed in Table 5-19. The "TNOW" delta type represents the difference in time. The "Attribute" delta type evaluates the difference in values based on entity properties and the "Expression" delta type evaluates the difference in values based on specified expressions. Once the delta checkbox in the "Delta" tab is checked (see Figure 5-38) and the delta type is selected, a parameter delta will be created automatically and shown in the accumulator expression field.

Table 5 - 19: Delta Type

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

5.4.1.2 Begin Collection Module

The *Begin Collection* module is used to trigger the beginning collection event (see Figure 5-40). Figure 5-41 shows the logic flow within the *Begin Collection* module.

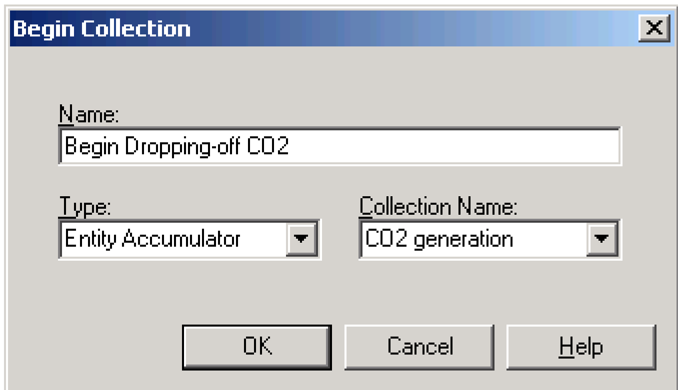


Figure 5 - 40: General Collection Modules: *Begin Collection* module

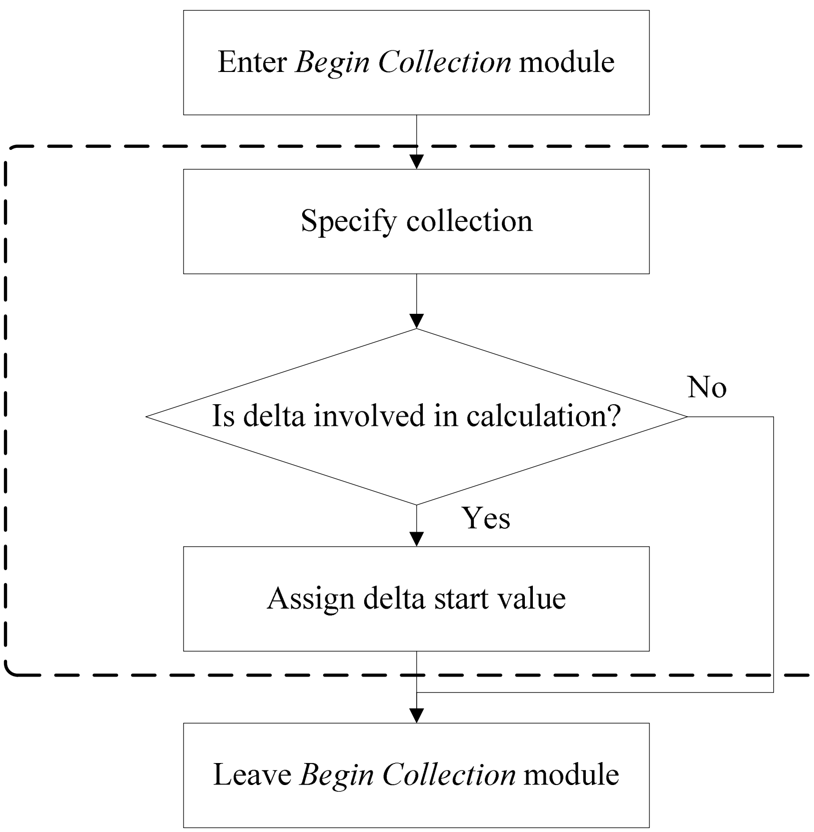


Figure 5 - 41: Flowchart of the logic for the *Begin Collection* Module

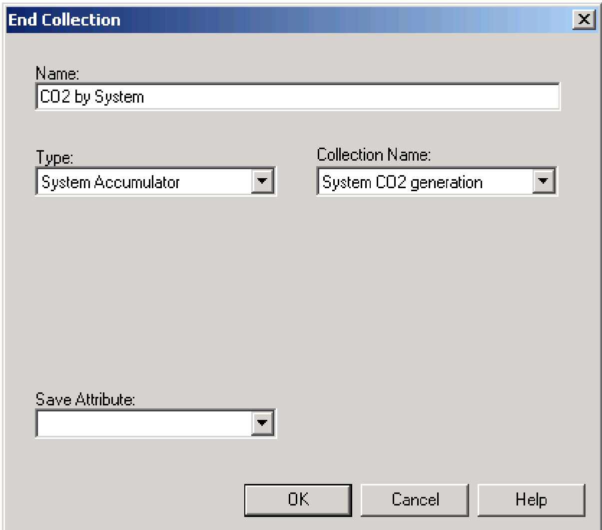
The *Begin Collection* module triggers the specified collection to start collecting. As pre-defined in the *Collection* data module, if delta is included in the corresponding calculation, the *Begin Collection* module assigns the initial or starting value of the delta depending on the delta type (see Figure 5-39). A "TNOW" 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 and an expression-based delta has an initial value of the current expression value. The algorithm for carrying out the logic flow is listed in Algorithm 5-11.

Algorithm 5 - 11: Pseudo-code for the *Begin Collection* Module

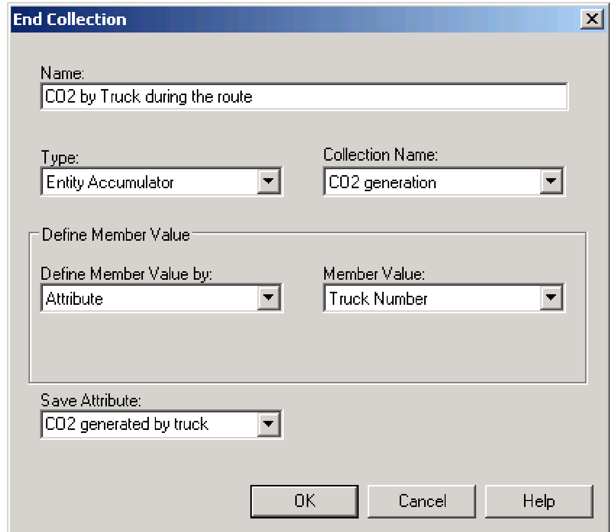
Step 1: Assign collection *i*;
Step 2: If delta is involved in calculation, go to Step 3;
  If no delta is involved in calculation, go to Step 4;
  Step 3: If delta type is time interval, set $Delta\_Start_i$ = current simulation time;
    If delta type is attribute, set $Delta\_Start_i$ = current attribute value;
    If delta type is expression, set $Delta\_Start_i$ = current expression value;
Step 4: Continue simulation.

5.4.1.3 End Collection Module

The *End Collection* module triggers the ending event of a pre-defined collection. The module also calculates and stores the corresponding environmental performance measure based on the accumulator expression (see Figure 5-42). The module first determines which collection to be ended. If the collection is a system accumulator (see Figure 5-42 (a)), the user specifies the collection that needs to be ended. The *End Collection* module ends the selected collection and performs the calculation based on the accumulator expression. The incremental value is also stored in the corresponding system variable.



(a)



(b)

Figure 5 - 42: General Collection Modules: *End Collection* module,
(a) System Accumulator, (b) Entity Accumulator

The *End Collection* module can also be applied to an entity accumulator (see Figure 5-42(b)). The user specifies which entity accumulator to end collection. The "Define Member Value" area specifies how to evaluate the "Member Value" specified in the *Collection* data module. Either by attribute or by expression, as shown in Figure 5-42(b), the *End Collection* module evaluates the current value of the attribute or expression listed in the "Member Value" drop-down menu. If the value matches one of the values pre-defined in the "Member Value" tab in the *Collection* data module, the entity triggers the end collection event and calculates the performance measure. Otherwise, a warning dialogue will be displayed to the user indicating that the passing entity cannot trigger the specified end collection event. Similar to system accumulator, the *End Collection* module calculates the entity accumulator based on the accumulator expression from the *Collection* data module. The incremental value is also stored in the corresponding system variable.

Last, the user can choose to save the incremental value as an attribute of the passing entity for both system accumulator and entity accumulator (see Figure 5-42). A flow chart of the

logic in the *End Collection* module is shown in Figure 5-43. The pseudo codes for executing the logic flow are listed in Algorithm 5-12.

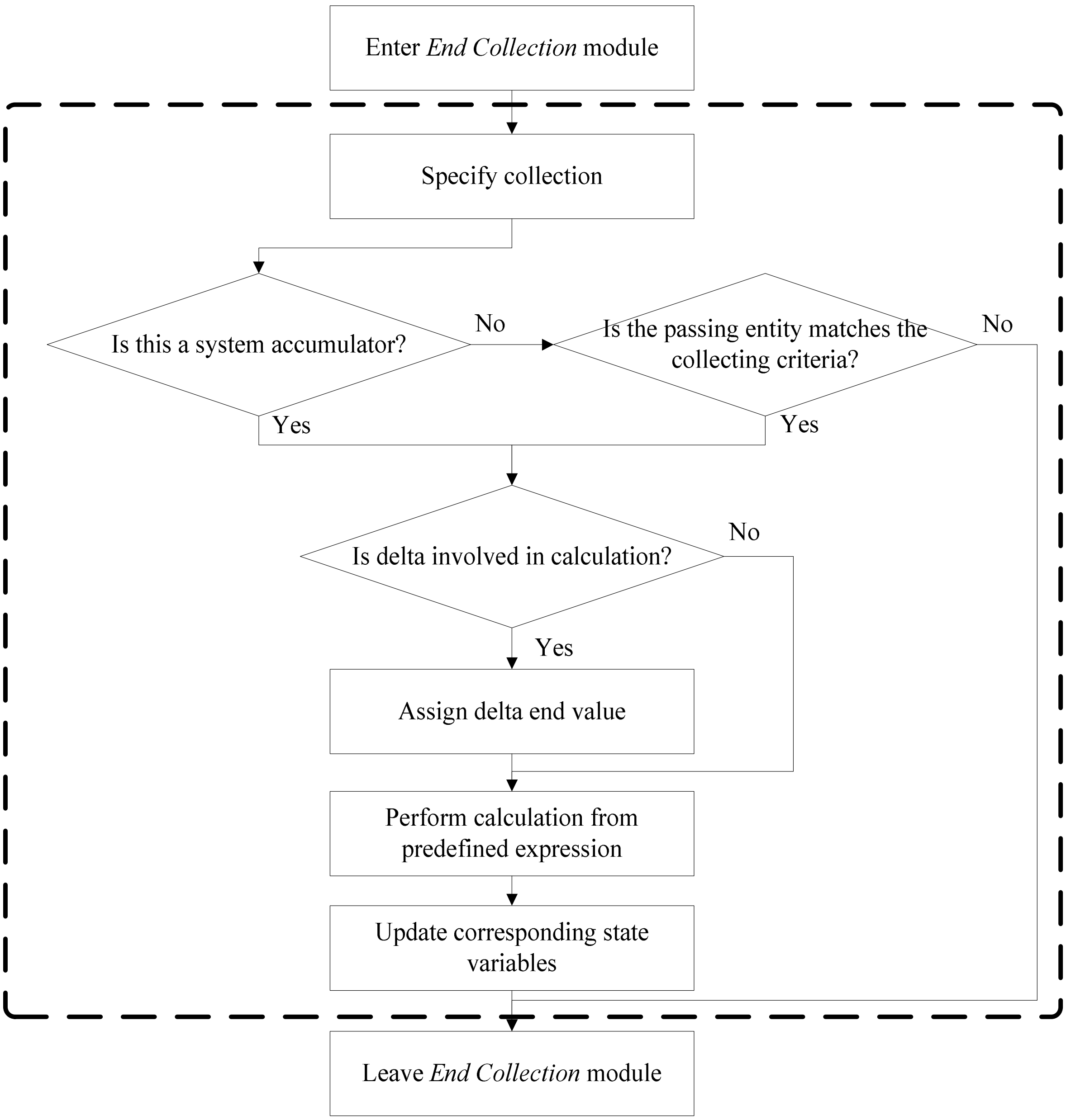


Figure 5 - 43: Flowchart of the logic for the *End Collection* Module

Algorithm 5 - 12: Pseudo-code for the *End Collection* Module

Step 1: Assign collection i;
Step 2: If collection is an entity accumulator, go to Step 3;
If collection is a system accumulator, go to Step 4;
Step 3: If passing entity $\in G_i$, go to Step 4;
If passing entity $\notin G_i$, go to Step 8;
Step 4: If delta is involved in calculation, go to Step 5;
If no delta is involved in calculation, go to Step 7;
Step 5: If delta type is time interval, set $Delta\_End_i$ = current simulation time;
If delta type is attribute, set $Delta\_End_i$ = current attribute value;
If delta type is expression, set $Delta\_End_i$ = current expression value;
Step 6: Set $Delta_i = Delta\_End_i - Delta\_Start_i$, go to Step 7;
Step 7: Set $Var_i = Var_i + Exp_i$;
Step 8: Continue simulation.

5.4.1.4 Remodel using the General Collection Modules

Example 5-4 of the transportation and logistics problem is modeled to demonstrate the usage of the general toolkit modules. All the system specifications and configurations remain the same as shown in Table 5-4. An overview of the simulation model using the general collection modules is displayed in Figure 5-44.

First the collections are defined with expressions of how to calculate each of the collections. In this particular simulation model, 8 trucks are created. 20% soya blend biodiesel is used and we are collecting the environmental performance measures of CO_2 and NO_x generation. CO_2 and NO_x emissions are collected by each individual truck and by the entire system of 8 trucks in total. The entities (the trucks) are created with truck information and route information assigned. Then the *Begin Collection* modules (labeled Begin CO_2 Collection, Begin NO_x Collection, Begin System CO_2 Collection, and Begin System NO_x Collection) are used to begin the collection of trucks and system CO_2 and NO_x generation. Then the trucks are sent to the customers. After the truck arrives at a customer location, the *End Collection* modules (labeled CO_2 by Truck during the rout, CO_2 by System, NO_x by Truck during the rout, and NO_x by System) are applied to perform the calculation of emissions generated along the route, and

update the state variables for trucks and the system. Another two sets of begin and end collection modules (labeled Begin Drop-off CO_2 and CO_2 by truck at Customer Location, for instance) are used to change and update the CO_2 and NO_x generation measures when the trucks are left idling at the customer's location. The trucks will visit all customers in the route following the same modeling logic. If the trucks have visited all customers assigned, the trucks return to the warehouse and report desired performance measures.

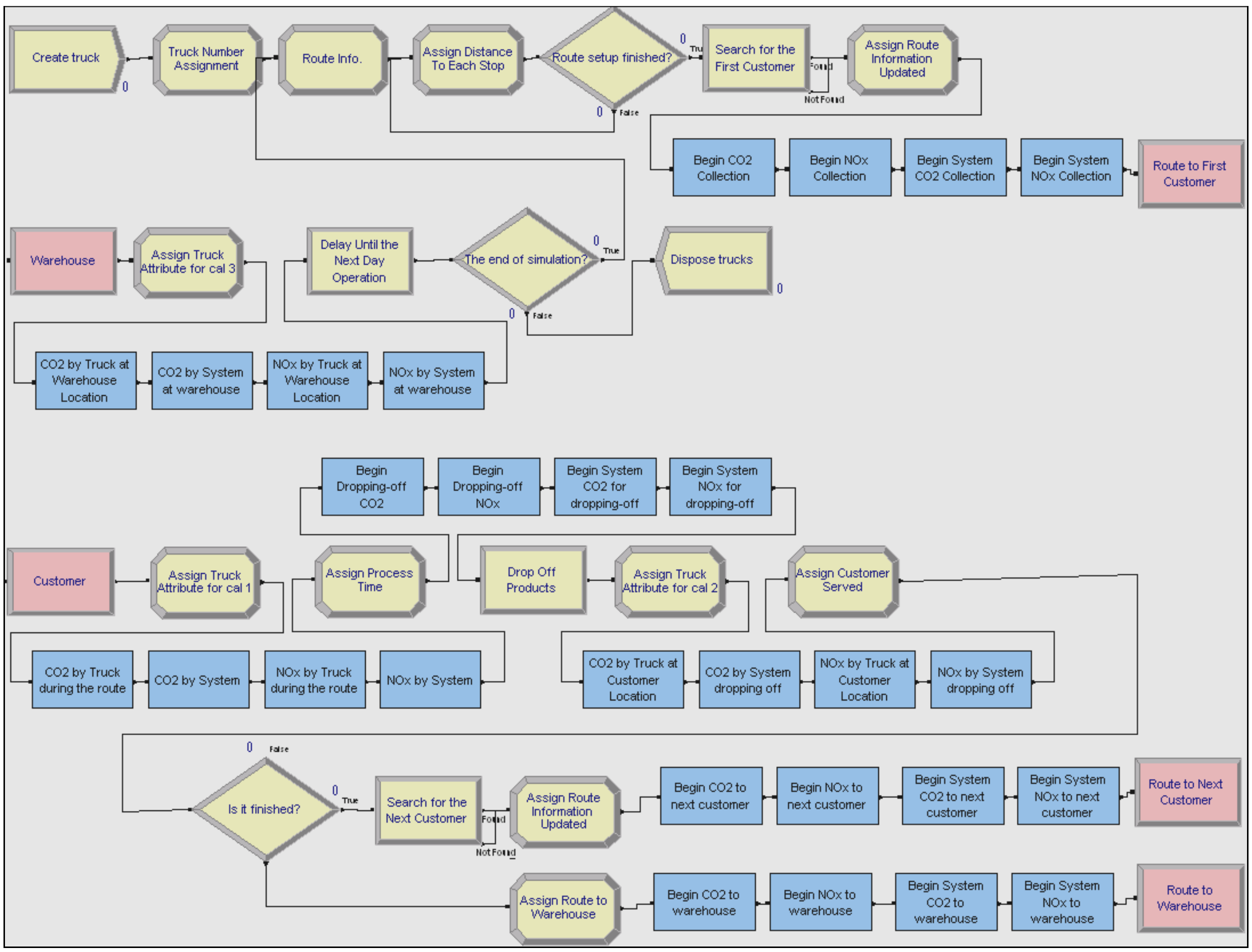


Figure 5 - 44: Example 5-4 Simulation Model using General Collection Modules

The model is set up and run for one period of five days simulation length. Table 5-20 shows a typical output report using general collection modules. The environmental performance

measures as predefined collections, including CO_2 and NO_x generation in this case, are collected and reported to the user.

Table 5 - 20: Simulation Results for Trucks using 20% Soya Blend Biodiesel over 5 Days

	CO_2 Emissions (gram)	**NOx Emissions (gram)**
Entity Accumulator		
Truck 1 Statistics	3.54E+06	24039
Truck 2 Statistics	2.26E+06	15314
Truck 3 Statistics	3.47E+06	23581
Truck 4 Statistics	2.31E+06	15668
Truck 5 Statistics	4.10E+06	27809
Truck 6 Statistics	1.99E+06	13522
Truck 7 Statistics	2.83E+06	19205
Truck 8 Statistics	2.45E+06	16611
System Accumulator		
System Statistics	2.29E+07	1.56E+05

5.4.2 Verification, Validation and Testing of the General Collection Modules

The general collection modules are debugged and are verified to work properly. The users are able to input their user-defined environmental performance measures and collect these performance measures throughout the simulation model using the general collection modules. At the end of the simulation, the modules are able to output the user-defined environmental performance measures directly to the user. In order to test the general collection modules that the modules are not affecting the system but collecting and reporting the user-defined performance measures, the original simulation model, which is also constructed in section 5.1.1, and the simulation model using the general collection modules are run for 100 replications of one week (5 days) period. A 95% confidence interval of the mean value of the system level CO_2 and NO_x is constructed for comparison in Table 5-18. The results are reasonably the same. A two-sample-t test of the means of system CO_2 and NO_x generation running from the modeling with and without the toolkit models is conducted using the Arena Output Analyzer. As a result, no

statically significant differences are found within the two models (see Figure 5-21). The general collection modules are tested and verified.

Table 5 - 21: Comparison of 95% Confidence Intervals on System CO2 and NOx Generation

	Original Model without Toolkit	**Model with Toolkit**	**Value Difference**	
	Average	**Average**	**95% CI Min Value**	**95% CI Max Value**
System CO_2 Emissions	2.2E+7	2.2E+7	-4.87E+5	5.54E+5
System NO_x Emissions	1.5E+5	1.5E+5	-3300	3760

5.4.3 Summary of the General Collection Modules

The general collection modules generalize the sustainability toolkit in such as 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.

6 CONCLUSIONS & RECOMMENDATION FOR FUTURE RESEARCH

The goal of this research was to develop a simulation-based sustainability toolkit that can aid in decision-making by having sustainability performance measures analyzed as easily as the traditional performance measures. The sustainability toolkit developed in this research can be used to model and analyze the environmental performance measures for transportation and logistics systems, industrial and manufacturing processes, and material handling systems. The conclusions and the recommendation for future work are presented in the next section.

6.1 Conclusions

In this research, the development methodology to create such a sustainability toolkit is established and applied to three major systems. Table 7-1 shows the systems studied and the corresponding environmental performance measures that are collected and reported using the sustainability toolkit for the three types of systems.

Table 6 - 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	
→ Industrial Coating Process	• Paint and Solvent Consumption • VOC Generation
→ Injection Molding Process	• Energy Consumption • CO_2, SO_2, NO_x, CH_4 Generation
→ Plastics Processing	• Energy Consumption • Water Usage • CO_2, SO_x, NO_x Generation
Material Handling Systems	• HC, CO, NO_x, PM Generation

The sustainability toolkit is implemented in ARENA simulation software. The toolkit can be used for simulation modeling and analysis along with other standard ARENA modules. The toolkit is able to capture the environmental performance measures as listed above, and keep updating them in the simulation model. Eventually, the sustainability toolkit reports the corresponding environmental performance measures at the end of the simulation. The environmental performance measures are collected and reported as efficiently as the traditional system performance measures.

Furthermore, a general collection toolkit has been developed to model systems of interest. Besides the three mainly discussed systems in this research, the general sustainability toolkit can be used to other systems and evaluate the sustainability factors and performance measures. The modeler has the flexibility of specifying user-defined environmental statistics but not losing the effectiveness for analyzing these statistics as a built-in integrated component of the toolkit. The environmental performance measures report is ready to be examined at the end of the simulation.

The sustainability toolkit has a user-friendly interface where the user can input collection parameters easily. The toolkit also uses a drag-and-drop modeling manner as other standard modules in ANRNA. From the prototype implementation of the toolkit, now simulation researchers and practitioners, who are interested in including sustainability and environmental impact performance measures as an integral part of the decision making process, could benefit from using the designed sustainability toolkit.

6.2 Recommendations 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. We list here some of the limitations of this research where future work can focus on. The limitations include:

1) The data and mathematical relationships for calculating the environmental performance measures are not generic. For each studied system in this research, the data and mathematical relationships for calculating the environmental performance measures are obtained from a particular study paper. Assumptions are made along with the used paper. It is not generic enough to cover all end-user needs. For instance, the default environmental performance measures in transportation and logistics modules are carbon monoxide (CO), carbon dioxide (CO_2), nitrogen oxides (NO_x) and total hydrocarbon (THC) from Manicom et al. (1993). Commonly studied air pollution, particulate matter (PM), is not included in the used study. It is also realized that the paper results are not able to represent the average amount of air pollutions generated by running trucks. The same limitations exist in the industrial and manufacturing process modules and the material handing modules.
2) The system-based toolkit modules can model three types of systems including the transportation and logistics systems, the industrial and manufacturing systems and the material handling systems. If the user needs to model and evaluate the environmental impact in other systems of interest, he/she has to use the general collection modules instead.
3) The sustainability toolkit modules are designed to evaluate the environmental impact only within a studied system. The social impact and the economic impact analysis are not included in the current toolkit modules.

4) The sustainability toolkit is implemented in particular simulation software ARENA. It requires basic ARENA modeling skills in order for the user to use the sustainability toolkit.

The following list provides some recommendations for future research that may further expand the usage of the sustainability toolkit:

1) *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.
2) *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.
3) *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 evaluate 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.

4) *Implement sustainability toolkit in general programming language SysML*: The sustainability toolkit developed in this research is built in specific simulation software (ARENA) which requires specific knowledge on operating the software itself. A general programming language (e.g. C++, Java) is more known by most modelers. If the sustainability toolkit is written and built using a general programming language, the toolkit can be adapted to off-the-shelf simulation software. In this case, the end user has the flexibility of choosing various simulation packages and easily loading the corresponding sustainability toolkit into the simulation package.

SysML (Systems Modeling Language), published in 2006, 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, the 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. Huang and McGinnis (2007) introduce the system simulation using SysML and create a translator to translate the SysML simulation model into eM-plant simulation software. McGinnis and Ustun (2009) illustrate an example of writing SysML simulation model into ARENA simulation software. Schonherr and Rose (2009) introduce an approach for automatic simulation

model generation using SysML and a translator translating SysML model to AnyLogic simulation software. Therefore, implementing the sustainability toolkit in SysML would be a good approach for future research and study.

REFERENCES

Ashby, M. F. (2009). *Materials and the Environment: Eco-informed Material Choice*. Butterworth-Heinemann, Oxford, UK.

Banks, J. (1998). Software for Simulation. In J. Banks (Ed.), *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*. John Wiley & Sons, New York.

Barth, M., & Todd, M. (1999). Simulation model performance analysis of a multiple station shared vehicle system. *Transportation Research Part C 7*, 237-259.

Blanco, J. A., Zavala, M. A., Imbert, J. B., & Castillo, F. J. (2005). Sustainability of forest management practices: Evaluation through a simulation model of nutrient cycling. *Forest Ecology and Management, 213*(1-3), 209-228.

Cortés, C. E., Vargas, L. S., & Corvalán, R. M. (2008). A simulation platform for computing energy consumption and emissions in transportation networks. *Transportation Research Part D: Transport and Environment, 13*(7), 413-427.

Costanza, R., d'Arge, R., de Groot, R., Farber, S., Grasso, M., Hannon, B., et al. (1997). The value of the world's ecosystem services and natural capital. *NATURE, 387*, 253-260.

Curran, M. A. (1996). *Environmental Life-Cycle Assessment*: McGraw-Hill, New York.

Davis, S. C., Diegel, S. W., & Boundy, R. G. (2009). *Transportation Energy Data Book: Edition 28*. Retrieved February 2011, from http://www.kerstech.com/PDFs/Transportation%20 Energy%20Data%20Book%202009.pdf.

Department of Energy (2006). *Cambridge Energy Research Associates' (CERA) 25th Executive Conference*. Houston, Texas. Retrieved July 2009, from http://www.energy.gov/news/3178.htm.

Environmental Protection Agency (2002) *Exhaust Emission Factors for Nonroad Engine Modeling - Spark-Ignition*. Retrieved February 2011, from http://www.epa.gov/OMS/models/nonrdmdl/p02015.pdf.

Environmental Protection Agency (2004). *Median Life, Annual Activity, and Load Factor Values for Nonroad Engine Emissions Modeling*. Retrieved February 2011, from http://www.epa.gov/otaq/models/nonrdmdl/nonrdmdl2004/420p04005.pdf.

Environmental Protection Agency (2006). *Greenhouse Gas Emissions from the U.S. Transportation Sector, 1990-2003*. Retrieved February 2011, from http://www.epa.gov/oms/climate/420r06003.pdf.

Energy Information Administration (2007). *Emissions of Greenhouse Gases Report*. Retrieved February 2011, from http://www.eia.doe.gov/oiaf/1605/ggrpt/carbon.html.

Energy Information Administration (2008). *International Energy Annual 2006*. Retrieved February 2011, from http://www.eia.doe.gov/pub/international/iealf/tablee1.xls.

Environmental Protection Agency (2008). *Municipal solid waste in the United States: 2007 facts and figures*. Retrieved February 2011, from http://www.epa.gov/waste/nonhaz/municipal/pubs/msw07-rpt.pdf.

Energy Information Administration (2009). *U.S. Carbon Dioxide Emissions from Energy Sources 2008 Flash Estimate*. Washington, DC. Retrieved February 2011, from http://www.eia.doe.gov/oiaf/1605/flash/pdf/flash.pdf.

Environmental Protection Agency (2010). *Exhaust Emission Factors for Nonroad Engine Modeling - Spark Ignition.* Retrieved February 2011, from http://www.epa.gov/OMS/models/nonrdmdl/nonrdmdl2010/420r10019.pdf.

Halima, I., & Srinivasan, R. (2008). Designing sustainable alternatives for batch operations using an intelligent simulation-optimization framework. *Chemical Engineering Research and Design, 86*(7), 809-822.

Heilala, J., Vatanen, S., Tonteri, H., Montonen, J., Lind, S., Johansson, B., and Stahre, J., (2008). Simulation-based Sustainable Manufacturing System Design, In *Proceedings of the 2008 Winter Simulation Conference*, S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds., Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, 1922-1930.

Hendrickson, C. T., Lave, L. B., & Matthews, H. S. (2006). *Environmental Life Cycle Assessment of Goods and Services: An Input-Output Approach*: RFF Press, Washington.

Henriksen, J. O., & Crain, R. C. (1998). *GPSSLH reference manual* (4 ed.). Annandale, Virginia: Wolverine Software Corporation.

Hoekstra, A. Y. (2006). *The Global Dimension of Water Governance: Nine Reasons for Global Arrangements in Order to Cope with Local Problems*. Retrieved February 2011, from http://www.waterfootprint.org/Reports/Report_20_Global_Water_Governance.pdf.

Holmberg, J., Lundqvist, U., Robèrt, K.-H., & Wackernagel, M. (1999). The ecological footprint from a systems perspective of sustainability. *International Journal of Sustainable Development and World Ecology, 6*.

Huang, E., Ramamurthy, R., & McGinnis, L. F., (2007). System and Simulation Modeling using SysML, In *Proceedings of the 2007 Winter Simulation Conference*, S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds., Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, 796-803.

Imagine That, Inc. (2006). *Extend User's Manual*. San Jose, California.

Intergovernmental Panel on Climate Change (2007). *Climate Change 2007 Synthesis Report*. Retrieved February 2011, from http://www.ipcc.ch/pdf/assessment-report/ar4/syr/ar4_syr_spm.pdf.

Jeon, C. M., & Amekudzi, A. (2005). Addressing sustainability in transportation systems: Definitions, indicators, and metrics. *Journal of Infrastructure Systems, 11*(1), 31-50.

Kelton, W. D., Sadowski, R. P., & Sturrock, D. T. (2006). *Simulation with Arena* (4 ed.). McGraw-Hill, New York.

Kuhl, M. E., & Zhou, X., (2009). Sustainability Toolkit for Simulation-based Logistics Decisions, In *Proceedings of the 2009 Winter Simulation Conference*, M. D.

Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls, eds., Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, 1466-1473.

Kurkovsky, A. (2004). *Simulation in the analysis of sustainable development: Goals, planning, and evaluation.* Paper presented at the Proceedings of the Fourth IASTED International Conference on Modelling, Simulation, and Optimization, Anaheim, CA, USA.

Law, A. M. (2007). *Simulation Modeling and Analysis*. McGraw-Hill, New York.

Lee, C. (2000). *The Impacts of Traffic Flow Control on CO_2 Emissions from Passenger Cars Using Dynamic Network Model.* Paper presented at the IEEE Intelligent Transportation Systems, Dearborn, MI, USA.

Malherbe, L., & Mandin, C. (2007). VOC emissions during outdoor ship painting and health-risk assessment. *Atmospheric Environment, 41*(30), 6322-6330.

Manicom, B., Green, C., & Goetz, W. (1993). *Methyl Soyate Evaluation of Various Diesel Blends In a DDC 6V-92 TA Engine*. Mississauga, Ontario: Ortech International. Retrieved February 2011, from http://www.biodiesel.org/resources/reportsdatabase/reports/tra/19930421_TRA-027.pdf.

Martins, A. A., Mata, T. M., Costa, C. A. V., & Sikdar, S. K. (2007). Framework for sustainability metrics. *Industrial and Engineering Chemistry Research, 46*(10), 2962-2973.

Matthews, H. S., Lave, L., & MacLean, H. (2002). Life cycle impact assessment: A challenge for risk analysts. *Risk Analysis, 22*(5), 853-860.

Rentz, O., Peters, N., Nunge, S. & Geldermann, J. (2002). *Best Available Techniques (BAT) for the Paint- and Adhesive Application in Germany, Volume I: Paint Application*. Retrieved February 2011, from http://ec.europa.eu/environment/air/pollutants/stationary/solvents/activities/pdf/d030_dfiu_ifare_adhesive.pdf.

Rockwell Automation (2005). *Arena User's Guide*. Sewickley, Pennsylvania.

Ruwanpura, J., Prado, B., & Hettiaratchi, P. (2003). *A computer simulation model to predict waste generation and promote sustainable construction practices at construction sites.* Paper presented at the Canadian Society for Civil Engineering - 31st Annual Conference: 2003 Building our Civilization, Jun 4-7 2003, Montreal, H3H 2R9, Canada.

Schriber, T. J., & Brunner, D. T. (1998). How Discrete-Event Simulation Software Works. In J. Banks (Ed.), *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*. John Wiley & Sons, New York.

Schriber, T. J., & Brunner, D. T., (2008). Inside Discrete-Event Simulation Software: How it Works and Why it Matters, In *Proceedings of the 2008 Winter Simulation Conference*, S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds., Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, 182-192.

Swain, J. J. (2007). New Frontiers in Simulation (Biennial survey of discrete-event simulation software). *OR/MS Today, 34*(5), 12.

Thiriez, A. (2004). *An Environmental Analysis of Injection Molding*. Masters Thesis at Massachussetts Institute of Technology, Cambridge, MA, USA.

Thiriez, A., & Gutowski, T. (2006). An Environmental Analysis of Injection Molding. In *Proceedings of the 2006 IEEE International Symposium*, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, 195-200.

United Nations General Assembly (1987). *Report of the World Commission on Environment and Development: Our Common Future*. Retrieved February 2011, from http://www.un-documents.net/wced-ocf.htm.

Wohlgemuth, V., Page, B., & Kreutzer, W. (2006a). Combining discrete event simulation and material flow analysis in a component-based approach to industrial environmental protection. *Environmental Modelling and Software, 21*(11), 1607-1617.

Wohlgemuth, V., Page, B., Kreutzer, W., Gromyko, O., & Heydt, L. (2006b). *Modelling Economic and Ecological Aspects of Inventory Management Strategies within a Component-based Material-flow Simulator*. Paper presented at the 20th International Conference on Informatics for Environmental Protection (EnviroInfo-2006).

World Resources Institute (1998). *World Resources 1998-99*: Oxford University Press, Oxford, UK.

Yu, R., Pan, M., Railsback, S. F., & Leung, P. (2009). *A Prototype Agent Based Fishery Management Model of Hawaii's Longline Fishery*. Paper presented at the 18th World IMACS / MODSIM Congress.

Zhang, Y., Yang, Z., & Yu, X. (2006). Measurement and evaluation of interactions in complex urban ecosystem. *Ecological Modelling, 196*(1-2), 77-89.

Zhou, X., & Kuhl, M. E., (2010). Design and Development of a Sustainability Toolkit for Simulation, In *Proceedings of the 2010 Winter Simulation Conference*, B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, eds., Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, 1601-1612.

APPENDICES

Appendix A. ARENA Template Development for Transportation and Logistics Modules

Appendix A-1 displays the transportation and logistics modules user interfaces. Appendix A-2 contains the logic flow which is carried out within the modules. Appendix A-3 shows the underline logic SIMAN codes of the modules. Details of the development and implementation of transportation and logistics modules can be found in section 5.1.

A-1. Module Dialogue Interface

Figure A-1 shows the dialogue interface for truck assignment module. The *Name* field and the *Transporter Specification* field are shown in the first dialogue. Potentially other types of transporters can be added to the transporter specification drop-down menu. If truck type is selected, a second dialogue shows up after clicking on the *Enter* bottom. In the second dialogue, the user can specify the truck information including the truck horsepower (*Truck Horsepower*), the truck efficiency (*MPG*), the average speed in miles per hour (*Average Speed*), and the fuel capacity in gallons (*Fuel Capacity*). Five types of fuel are provided to the user (*Fuel Type*). The checkbox in front of each emission type is checked if the user wants to collect the emission measurement. Leaving it unchecked will result in not collecting the corresponding emission measure and deactivates the *Emission Coefficient* field. Default values are provided for the emission coefficients. User-specified emission coefficient values can also be entered.

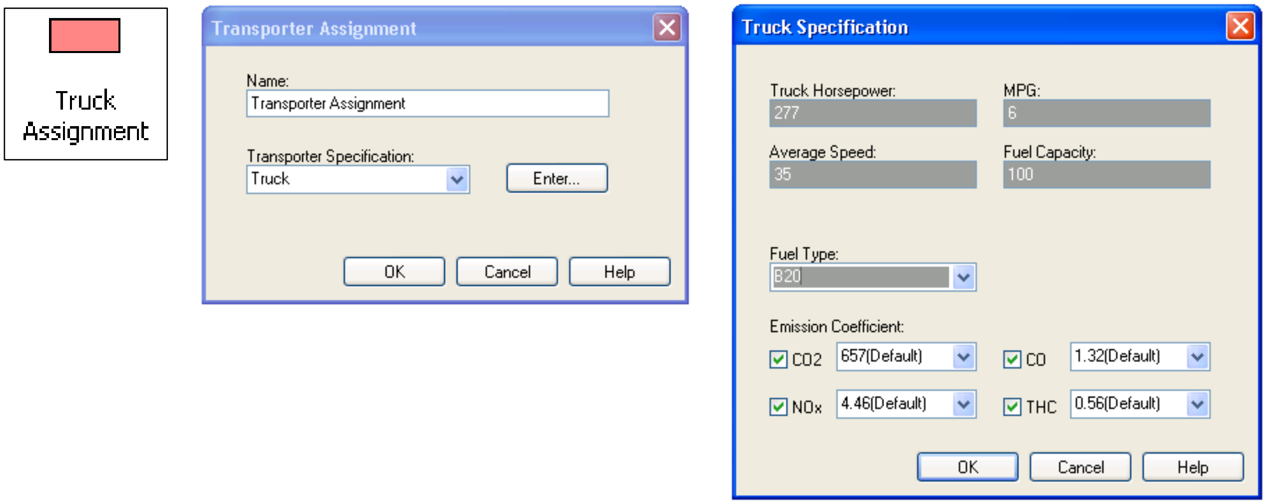


Figure A - 1: Truck Assignment Module Interface

Figure A-2 shows the dialogue interface for route with emissions module. The interface of the route with emission module is similar to a standard route module where route information is specified either by routing distance (*Distance*) or routing time (*Route time*). The destination station is specified using the *Destination Type* and *Station Name* fields.

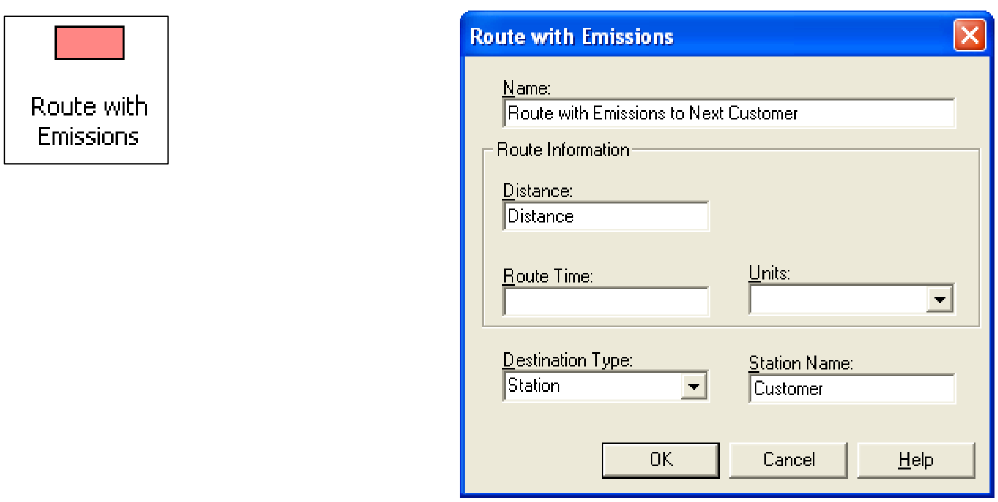


Figure A - 2: Route with Emissions Module Interface

Figure A-3 depicts the dialogue interface for station with emissions module. The *Name* and *Station Name* are used to specify the module name and the station name. Similar to the standard station module, the *Report Statistics* checkbox checks to return the station statistics. In

addition, the station with emissions module can specify the probability for trucks to be left idling at the station in the *Probability for Truck Idling* field.

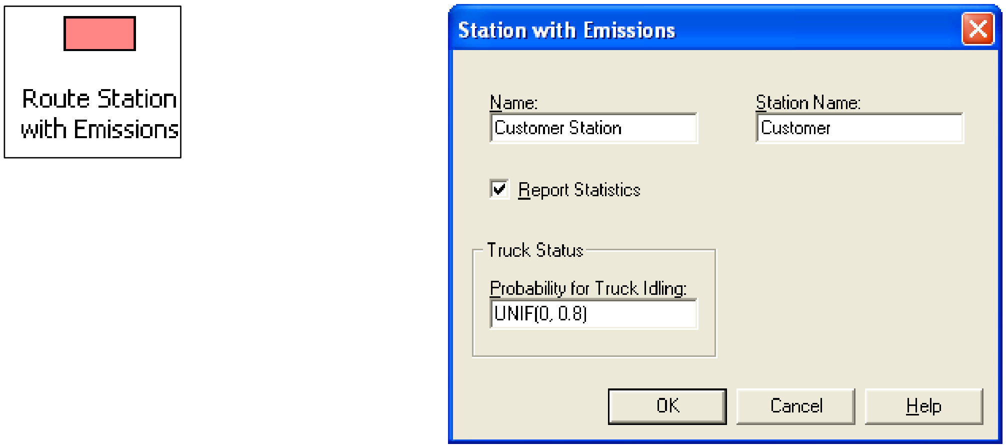


Figure A - 3: Station with Emission Module Interface

A-2. Module Logic Window

Truck Assignment Module

Figure A-4 shows the truck assignment module logic view implemented in ARENA simulation software. The lower portion of the logic view assigns truck attributes and emission collections information. The upper portion of the logic view generates the report automatically at the end of the simulation. Figures A-5 and A-6 depict the flow charts view of the truck assignment logic and report generation logic. The truck entity enters the module. The truck information including the truck HP, efficiency (MPG), average speed and truck number is assigned first. The fuel type is assigned. If the users choose to collect the emission types of CO, CO_2, NOx, and THC, corresponding collecting variables are set to be "1" and the emission coefficients are assigned. The truck entity then leaves the truck assignment module.

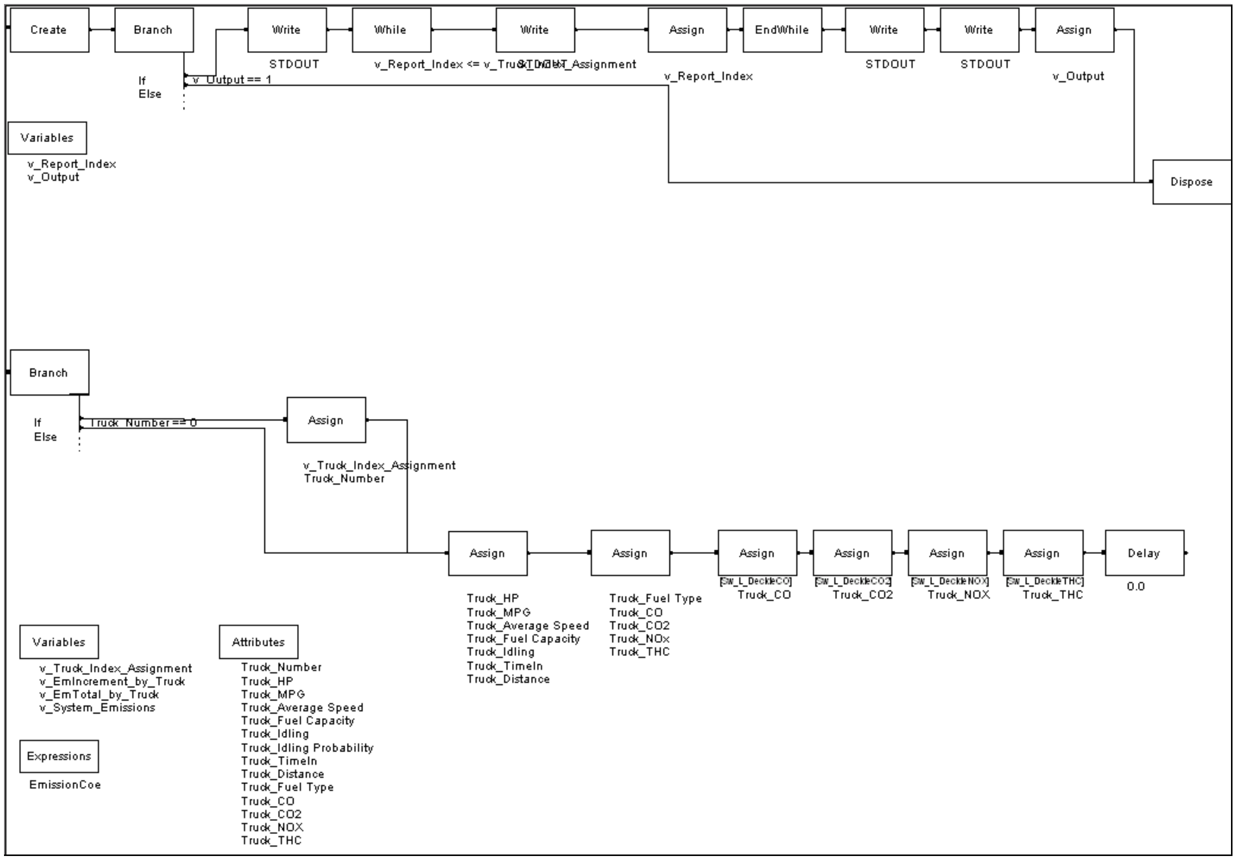


Figure A - 4: Truck Assignment Module ARENA Simulation Logic Overview

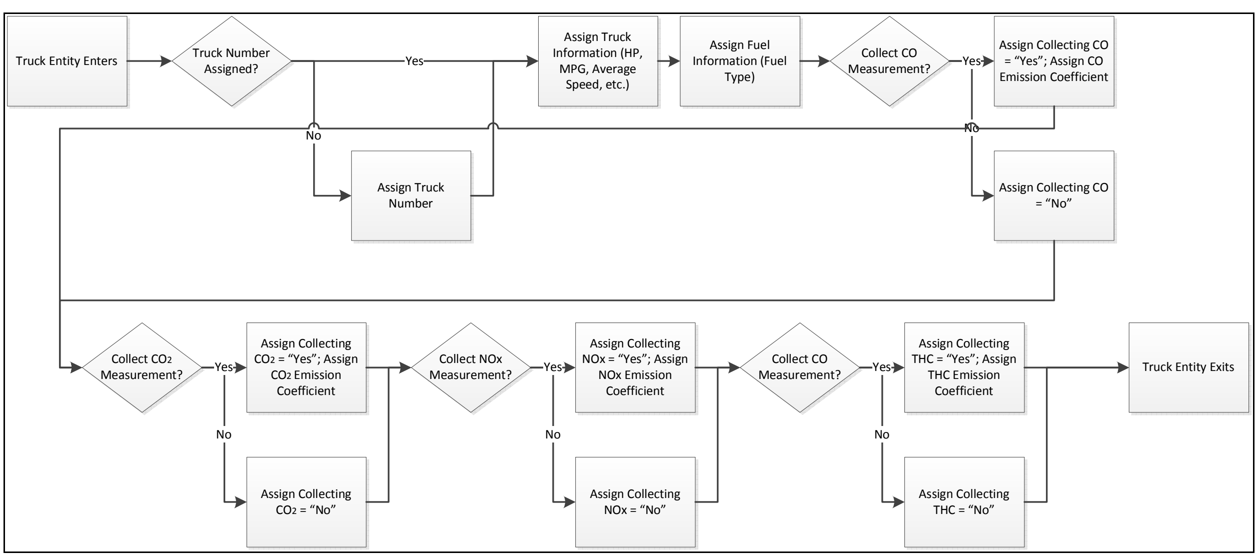


Figure A - 5: Truck Assignment Module Flow Chart View

In the report generation flow chart view as shown in Figure A-6, a control entity will be generated at the end of the simulation. According to the emission collecting variables' value (1 means the emission measure is collected), the emission measures are reported by each individual truck and by the entire simulation system. The control entity is disposed eventually.

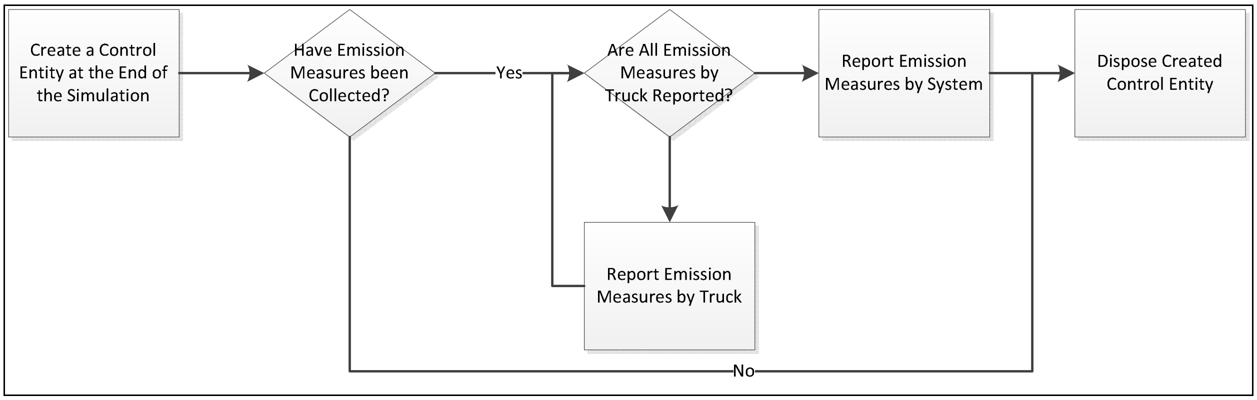


Figure A - 6: Truck Assignment Module Report Generation Flow Chart View

Route with Emission Module

Figure A-7 shows the route with emission module logic overview implemented using ARENA standard modeling modules. Figure A-8 shows the route with emission module logic flow chart view. The truck entity enters the route with emission module. The traveling distance will be first assigned to the truck entity. The logic will then check if the truck is left idling while remaining at the current station. If the truck is left idling, emission generation measures will be updated. If the truck is turned off, emission generation measures will not be updated. The truck then leaves the route with emission module and is routed to the next station as specified in the route with emission user interface.

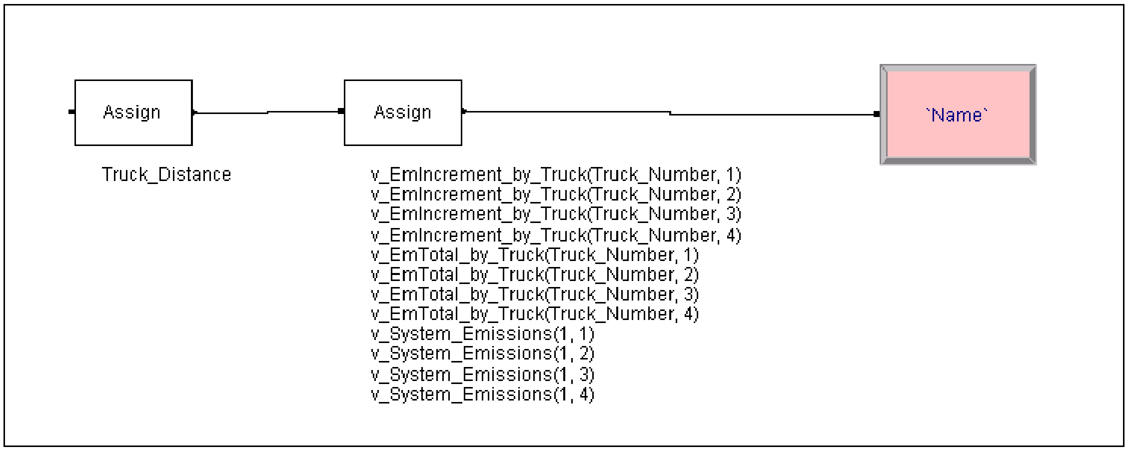


Figure A - 7: Route with Emission Module ARENA Simulation Logic Overview

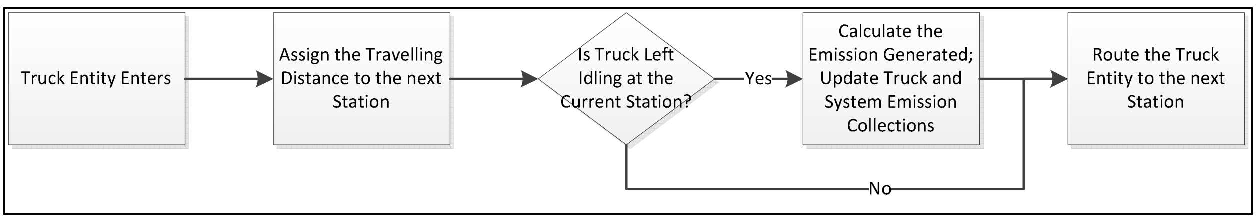


Figure A - 8: Route with Emission Module Flow Chart View

Route Station with Emission Module

The logic overview of the route station with emission module using ARENA simulation standard modules is shown in Figure A-9. Figure A-10 demonstrates the logic flow chart view of the route station with emission module. When the truck enters the station, the truck assigns the idling property first (value of 1 if the truck is left idling). The emission generation measures are then increased by the amount of emission generated along the route. The emission generation measures are updated by each individual truck and by the whole system.

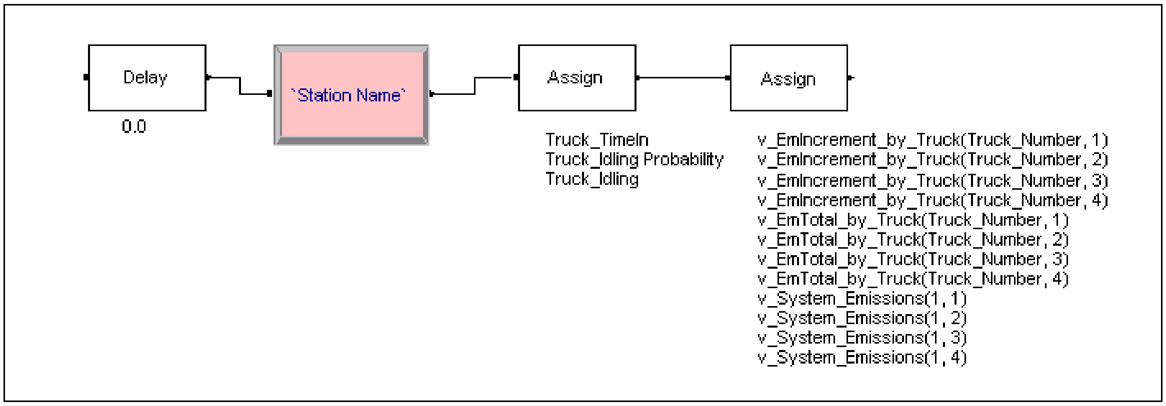


Figure A - 9: Route Station with Emission Module ARENA Simulation Logic Overview

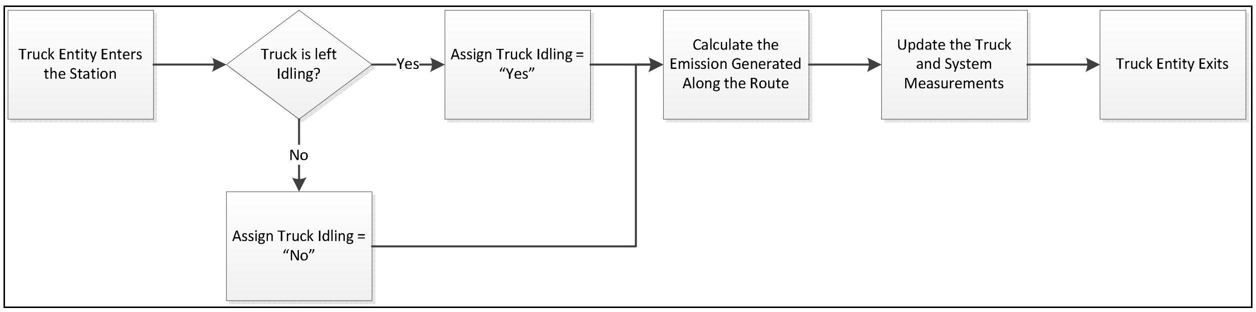


Figure A - 10: Route Station with Emission Module Flow Chart View

A-3. Module Underline Logic SIMAN Code

The following section lists the simulation SIMAN codes for the truck assignment module, the route with emission module and the route station with emission module.

Truck Assignment module

```
0$              CREATE,         1,TFIN:,1:NEXT(10$);

10$             BRANCH,         1:
                                If,v_Output == 1,1$,No:
                                Else,6$,No;
1$              WRITE,          STDOUT,"\n%s\n%s\n":
                                "---------------------------------------------------------
                                ---------------",
                                "The report of environmental performance measures are
                                listed as follows:";
2$              WHILE:          v_Report_Index <= v_Truck_Index_Assignment;
3$              WRITE,          STDOUT,
                                "\nTruck %i CO2 Emission: %6.2g\nTruck %i CO
                                Emission: %6.2g\nTruck %i NOX Emission: %6.2g\nTruck %i
                                THC Emission: %6.2g\n":
                                v_Report_Index,
                                v_EmTotal_by_Truck(v_Report_Index, 1),
                                v_Report_Index,
```

```
                            v_EmTotal_by_Truck(v_Report_Index, 2),
                            v_Report_Index,
                            v_EmTotal_by_Truck(v_Report_Index, 3),
                            v_Report_Index,
                            v_EmTotal_by_Truck(v_Report_Index, 4);
5$             ASSIGN:      v_Report_Index=v_Report_Index + 1;
4$             ENDWHILE;
7$             WRITE,       STDOUT,
                            "\nSystem CO2 Emission: %6.2g\nSystem CO
                            Emission: %6.2g\nSystem NOX Emission: %6.2g\nSystem THC
                            Emission: %6.2g\n":
                            v_System_Emissions(1,1),
                            v_System_Emissions(1,2),
                            v_System_Emissions(1,3),
                            v_System_Emissions(1,4);
8$             WRITE,       STDOUT,"\n%s\n":
                            "---------------------------------------------------------
                            ---------------";
9$             ASSIGN:      v_Output=0;
6$             DISPOSE:     No;

19$            BRANCH,      1:
                            If,Truck_Number == 0,18$,No:
                            Else,16$,No;
18$            ASSIGN:      v_Truck_Index_Assignment=v_Truck_Index_Assignment + 1:
                            Truck_Number=v_Truck_Index_Assignment;
16$            ASSIGN:      Truck_HP:
                            Truck_MPG:
                            Truck_Average Speed:
                            Truck_Fuel Capacity:
                            Truck_Idling=0:
                            Truck_TimeIn=TNOW:
                            Truck_Distance=0;
17$            ASSIGN:      Truck_Fuel Type:
                            Truck_CO=0:
                            Truck_CO2=0:
                            Truck_NOX=0:
                            Truck_THC=0;
11$            ASSIGN:      Truck_CO=1;
12$            ASSIGN:      Truck_CO2=1;
13$            ASSIGN:      Truck_NOX=1;
14$            ASSIGN:      Truck_THC=1;
15$            DELAY:       0.0,,Other;
```

Route with Emissions module

```
2$             ASSIGN:      Truck_Distance;
1$             ASSIGN:      v_EmIncrement_by_Truck(Truck_Number, 1)=
                            Truck_CO2*Truck_Idling * Truck_HP * (TNOW-Truck_TimeIn) *
EmissionCoe(Truck_Fuel Type,1):
                            v_EmIncrement_by_Truck(Truck_Number, 2)=
                            Truck_CO*Truck_Idling * Truck_HP * (TNOW-Truck_TimeIn) *
EmissionCoe(Truck_Fuel Type,2):
                            v_EmIncrement_by_Truck(Truck_Number, 3)=
                            Truck_NOX*Truck_Idling * Truck_HP * (TNOW-Truck_TimeIn) *
EmissionCoe(Truck_Fuel Type,3):
                            v_EmIncrement_by_Truck(Truck_Number, 4)=
                            Truck_THC*Truck_Idling * Truck_HP * (TNOW-Truck_TimeIn) *
EmissionCoe(Truck_Fuel Type,4):
                            v_EmTotal_by_Truck(Truck_Number, 1)=
                            v_EmTotal_by_Truck(Truck_Number, 1) +
v_EmIncrement_by_Truck(Truck_Number, 1):
```

```
                             v_EmTotal_by_Truck(Truck_Number, 2)=
                             v_EmTotal_by_Truck(Truck_Number, 2) +
v_EmIncrement_by_Truck(Truck_Number, 2):
                             v_EmTotal_by_Truck(Truck_Number, 3)=
                             v_EmTotal_by_Truck(Truck_Number, 3) +
v_EmIncrement_by_Truck(Truck_Number, 3):
                             v_EmTotal_by_Truck(Truck_Number, 4)=
                             v_EmTotal_by_Truck(Truck_Number, 4) +
v_EmIncrement_by_Truck(Truck_Number, 4):
                             v_System_Emissions(1, 1)=v_System_Emissions(1, 1) +
v_EmIncrement_by_Truck(Truck_Number, 1):
                             v_System_Emissions(1, 2)=v_System_Emissions(1, 2) +
v_EmIncrement_by_Truck(Truck_Number, 2):
                             v_System_Emissions(1, 3)=v_System_Emissions(1, 3) +
v_EmIncrement_by_Truck(Truck_Number, 3):
                             v_System_Emissions(1, 4)=v_System_Emissions(1, 4) +
v_EmIncrement_by_Truck(Truck_Number, 4)
                             :NEXT(0$);

0$            ROUTE:         Truck_Distance/Truck_Average Speed,Station 1;
```

Route Station with Emissions module

```
3$            DELAY:         0.0,,Other:NEXT(0$);
0$            STATION,       Station 1;
6$            DELAY:         0.0,,VA:NEXT(1$);

1$            ASSIGN:        Truck_TimeIn=TNOW:
                             Truck_Idling Probability:
                             Truck_Idling=DISC(Truck_Idling Probability, 1, 1.0, 0);
2$            ASSIGN:        v_EmIncrement_by_Truck(Truck_Number, 1)=
                             Truck_CO2 * Truck_HP * (Truck_Distance/Truck_Average
Speed) * EmissionCoe(Truck_Fuel Type,1):
                             v_EmIncrement_by_Truck(Truck_Number, 2)=
                             Truck_CO * Truck_HP * (Truck_Distance/Truck_Average
Speed) * EmissionCoe(Truck_Fuel Type,2):
                             v_EmIncrement_by_Truck(Truck_Number, 3)=
                             Truck_NOX * Truck_HP * (Truck_Distance/Truck_Average
Speed) * EmissionCoe(Truck_Fuel Type,3):
                             v_EmIncrement_by_Truck(Truck_Number, 4)=
                             Truck_THC * Truck_HP * (Truck_Distance/Truck_Average
Speed) * EmissionCoe(Truck_Fuel Type,4):
                             v_EmTotal_by_Truck(Truck_Number, 1)=
                             v_EmTotal_by_Truck(Truck_Number, 1) +
v_EmIncrement_by_Truck(Truck_Number, 1):
                             v_EmTotal_by_Truck(Truck_Number, 2)=
                             v_EmTotal_by_Truck(Truck_Number, 2) +
v_EmIncrement_by_Truck(Truck_Number, 2):
                             v_EmTotal_by_Truck(Truck_Number, 3)=
                             v_EmTotal_by_Truck(Truck_Number, 3) +
v_EmIncrement_by_Truck(Truck_Number, 3):
                             v_EmTotal_by_Truck(Truck_Number, 4)=
                             v_EmTotal_by_Truck(Truck_Number, 4) +
v_EmIncrement_by_Truck(Truck_Number, 4):
                             v_System_Emissions(1, 1)=v_System_Emissions(1, 1) +
v_EmIncrement_by_Truck(Truck_Number, 1):
                             v_System_Emissions(1, 2)=v_System_Emissions(1, 2) +
v_EmIncrement_by_Truck(Truck_Number, 2):
                             v_System_Emissions(1, 3)=v_System_Emissions(1, 3) +
v_EmIncrement_by_Truck(Truck_Number, 3):
                             v_System_Emissions(1, 4)=v_System_Emissions(1, 4) +
v_EmIncrement_by_Truck(Truck_Number, 4);
```

Appendix B. ARENA Template Development for Industrial Processes Module

Appendix B-1 displays the process with emissions module user interface. Appendix B-2 contains the logic flow which is carried out within the modules. Appendix B-3 shows the underline logic SIMAN codes of the modules. Details of the development and implementation of industrial and manufacturing modules can be found in section 5.2.

B-1. Module Dialogue Interface

Figure B-1 shows the industrial coating process user interface of the process with emissions module. In the *Specification* drop-down menu, the user selects the coating process type from standard pre-defined coating process and user-specified coating process. If the standard specification is selected, the left portion of the dialogue interface is activated while the right portion is deactivated. The user then enters the pre-defined coating parameters including the *Paint Type, Solvent Ratio, Surface Area,* and *Coating Efficiency.* The *Solvent Ratio* drop-down menu has some pre-defined value depending on the *Paint Type* selected (water-based or solvent-based). The *Surface Area* has pre-defined area units to choose from. The *Coating Efficiency* can be any number less than 1, or an expression as shown in Figure B-1. The user needs to choose the interested environmental measures to collect using the checkboxes. The pre-defined environmental measures include the *Paint Consumption*, the *Solvent Consumption* and the *VOC* generation. The *Collect Additional Statistics* commend button is used to collect additional measures besides the pre-defined three measures. By clicking the commend button, a second dialogue interface shows up (Figure B-2). The user can add additional statistics by attribute or by variable.

If a user-specified coating process is selected, the right portion of the dialogue becomes available to enter the user-defined measures. The user can add or edit the measures by clicking the *Add* and the *Edit* button.

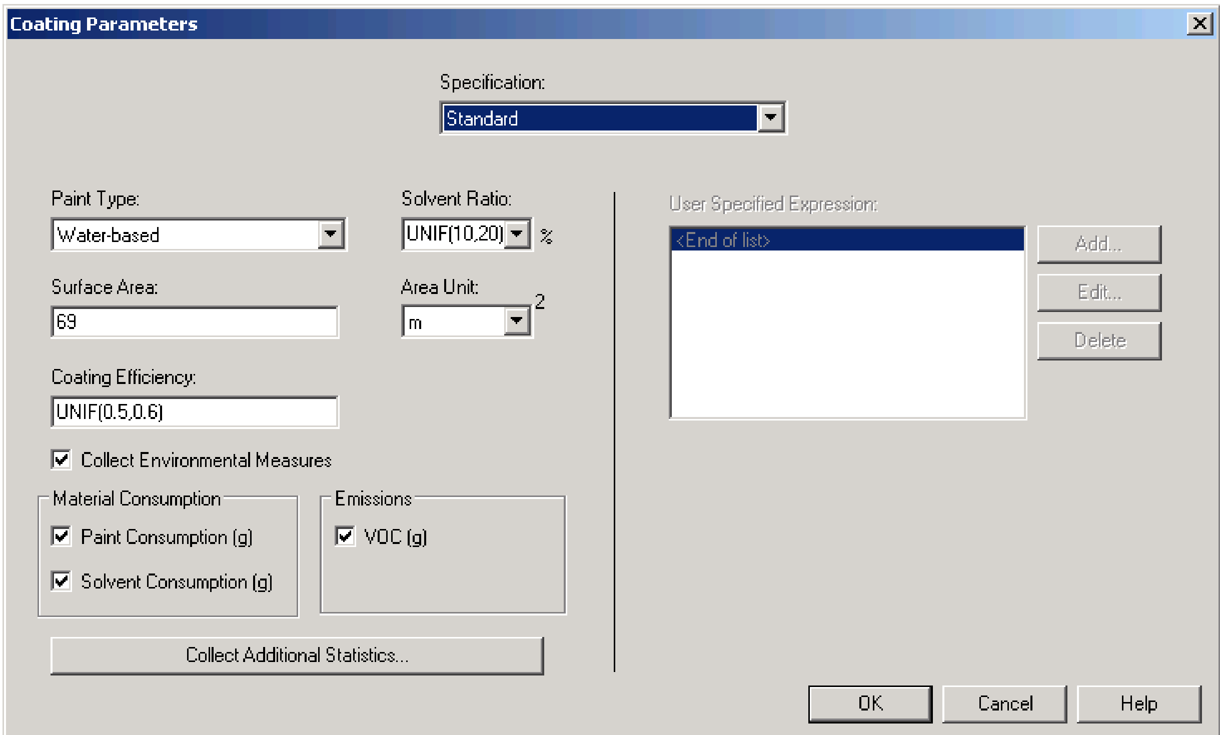


Figure B - 1: Industrial Coating Process Module Interface

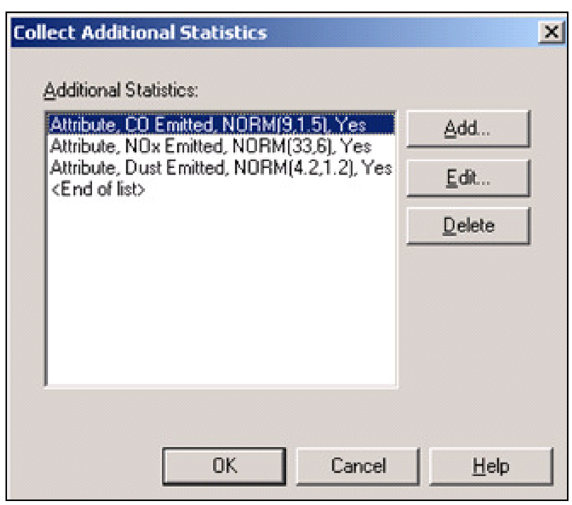


Figure B - 2: Collect Additional Statistics User Interface

The injection molding process user interface of the process with emissions module is shown in Figure B-3. Similar to the industrial coating process user interface, a standard injection molding process or a user-specified injection molding process needs to be selected in the *Specification* drop-down menu. For a standard injection molding process, the left portion of the dialogue is activated. The user chooses the *Injection Molding Machine* type among the hydraulic machine, all-electric machine and hybrid machine. The shot weight of the input polymer is entered in the *Input Polymer* field. The pre-defined environmental measures for injection molding process include energy consumption, CO_2 generation, NO_x generation, SO_2 generation and CH_4 generation. Environmental measures are collected and reported at the end of the simulation with the checkboxes of *Energy Consumption,* CO_2*,* NO_x*, and* SO_2 being checked. The standard injection molding process also provides a *Collect Additional Statistics* option. The right portion of the dialogue becomes available when a user-specified injection molding process is chosen.

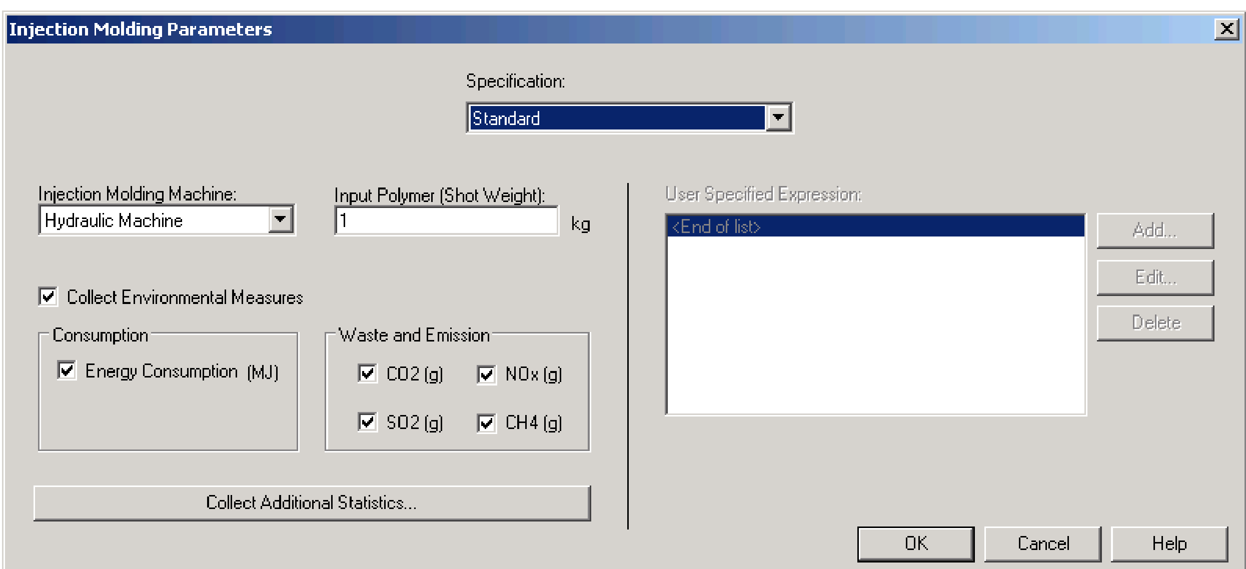


Figure B - 3: Injection Molding Process Module Interface

Figure B-4 depicts the plastics processing user interface of the process with emissions module. In a standard plastics processing process, the user chooses the plastics types from *Thermoplastics* and *Thermosetting Polymers*. The *Thermoplastics* drop-down menu becomes available when the plastics type is thermoplastics and the user can further choose from polypropylene and polyvinyl chloride. When the plastics type is thermosetting polymers, the *Thermosets* drop-down menu is activated and the user can choose the polyimide or polyurethane to process. The pre-defined environmental performance measures include the *Embodied Energy*, *Water Usage, CO_2 Generation, NO_x Generation* and *SO_x Generation.* The standard plastics processing process also provides a *Collect Additional Statistics* option. The right portion of the dialogue becomes available when a user-specified plastics processing is chosen.

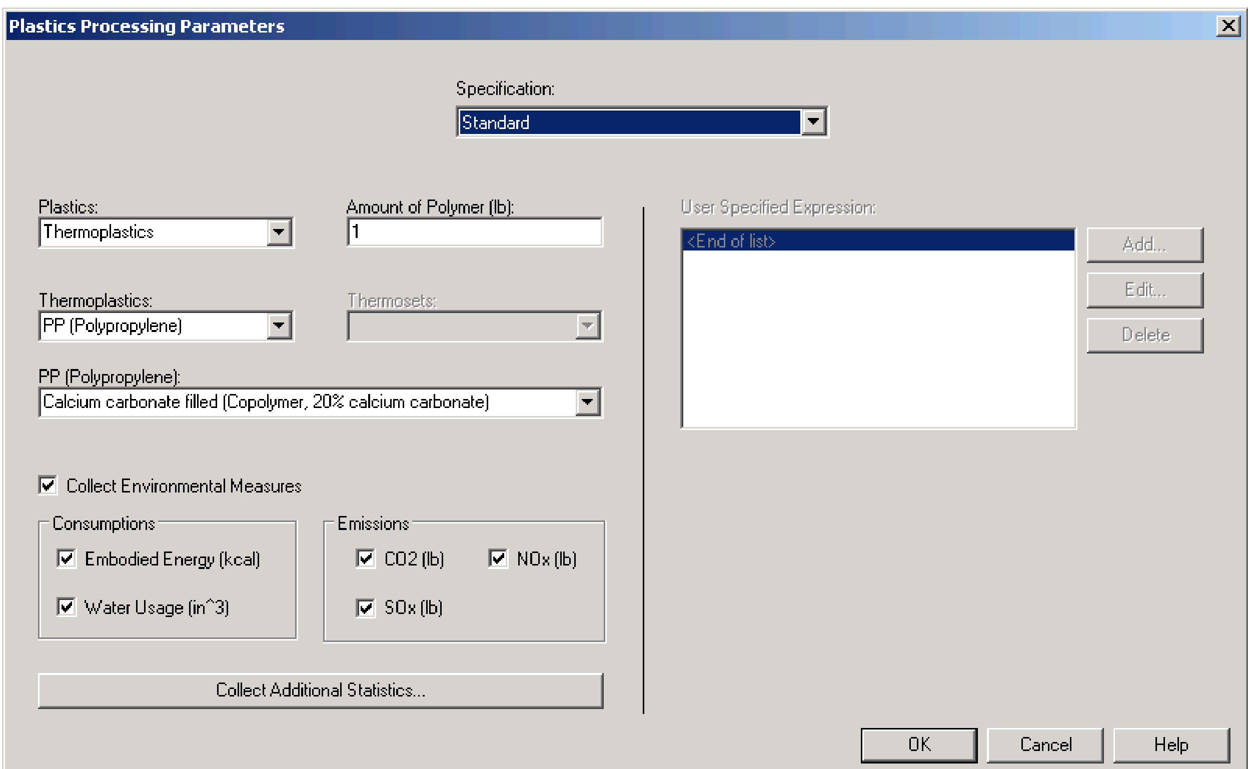


Figure B - 4: Plastics Processing Module Interface

B-2. Module Logic Window

Process with Emissions Module Overview

Figure B-5 shows the process with emissions module logic overview implemented in ARENA simulation software. The underline logic of the process with emissions module is constructed using ARENA standard modules and sub-models. Figure B-6 depicts the flow chart view of the process with emissions logic. The entity assigns the standard process parameters first including the process time and resources. Depending on the user input, the entity enters sub-process logic including industrial coating process, injection molding process, plastics process and user-defined process. The sub-process logic is modeled using sub-models.

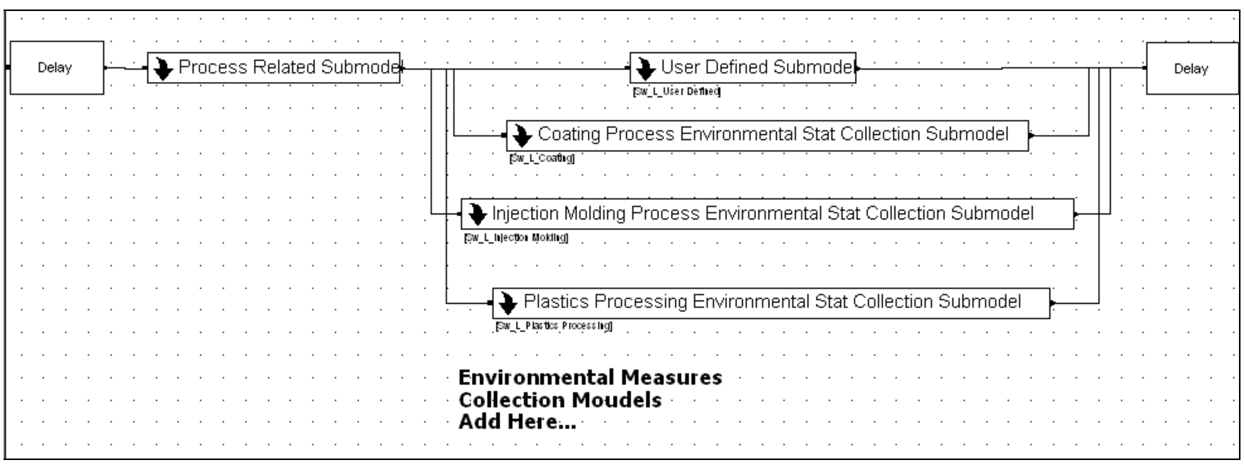


Figure B - 5: Process with Emissions Module ARENA Simulation Logic Overview

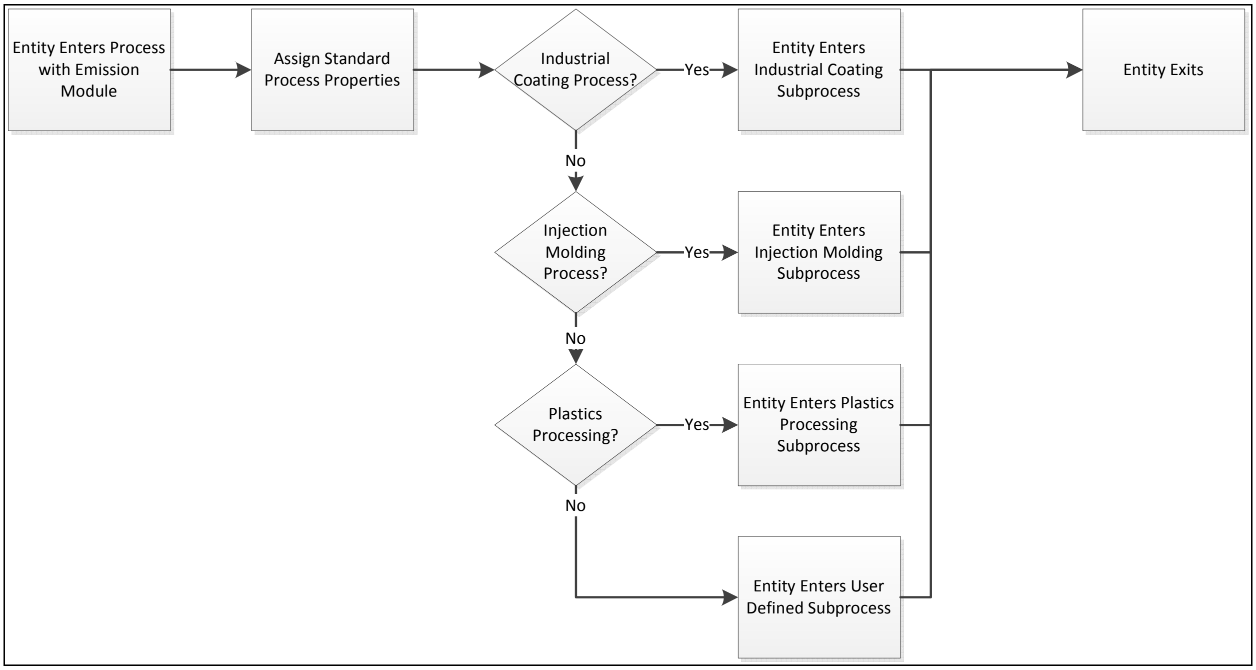


Figure B - 6: Process with Emission Module General Flow Chart View

Industrial Coating Process Sub-Model

Figure B-7 shows the industrial coating process ARENA simulation sub-model logic view. Figure B-8 shows the corresponding flow chart view of the logic. When the entity enters the coating process sub-model, it first checks if this is a standard coating process. A user-specified coating process will calculate and update the user-specified statistics and then the entity exits the sub-model. A standard coating process will first determine the paint type. The solvent ratio and coating parameters including the surface area and efficiency are assigned. Pre-defined environmental performance measures are updated. After collecting the standard measures, the entity checks if there are additional user-specified statistics. The additional user-specified statistics are collected and updated and the entity exits the coating process sub-model.

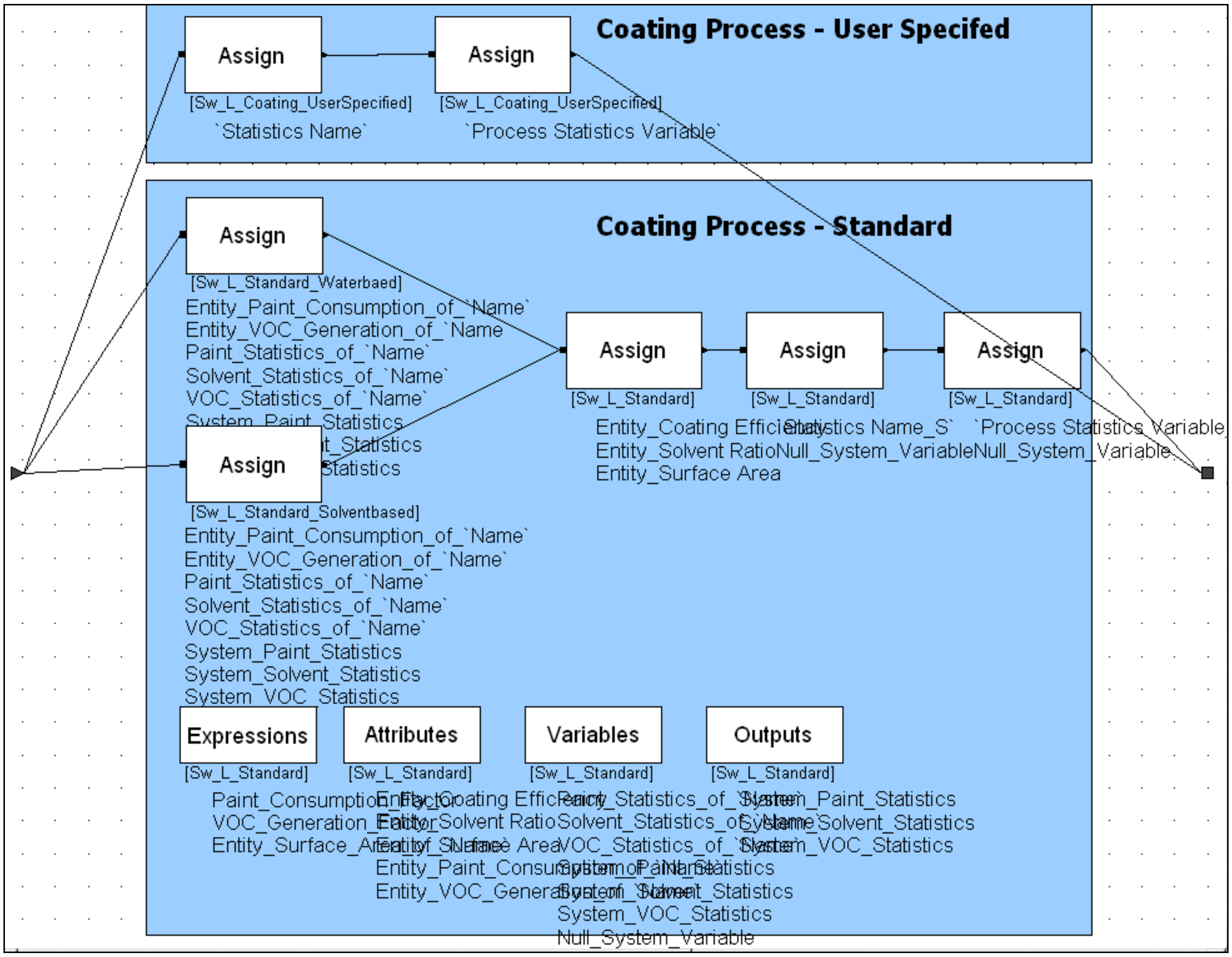


Figure B - 7: Industrial Coating Process ARENA Simulation Sub-Model Logic Overview

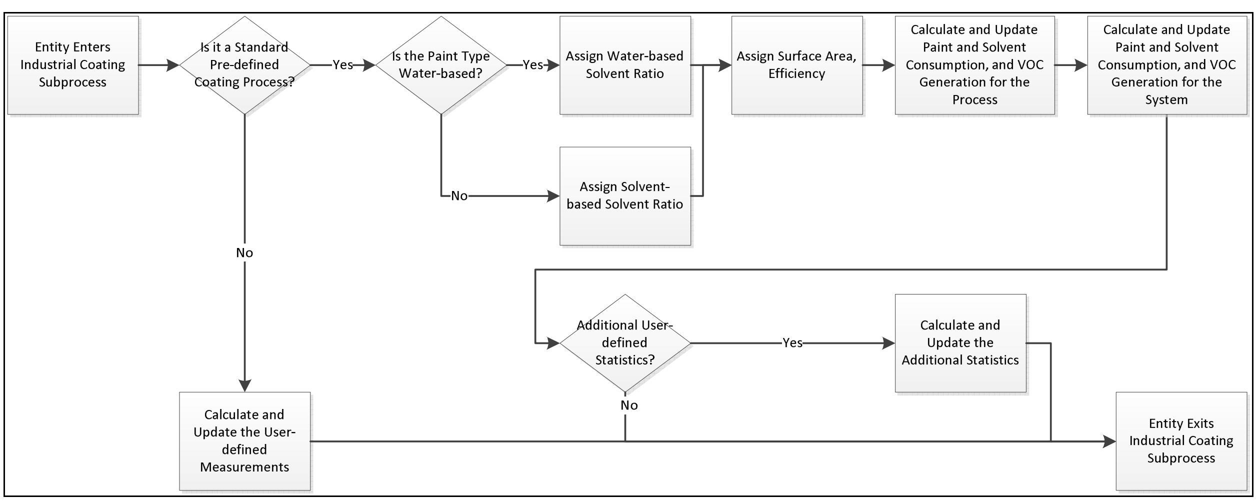


Figure B - 8: Industrial Coating Process Sub-Model Flow Chart View

Injection Molding Process Sub-Model

The injection molding process sub-model logic modeled in ARENA is shown in Figure B-9. Figure B-10 depicts the corresponding flow chart view of the logic.

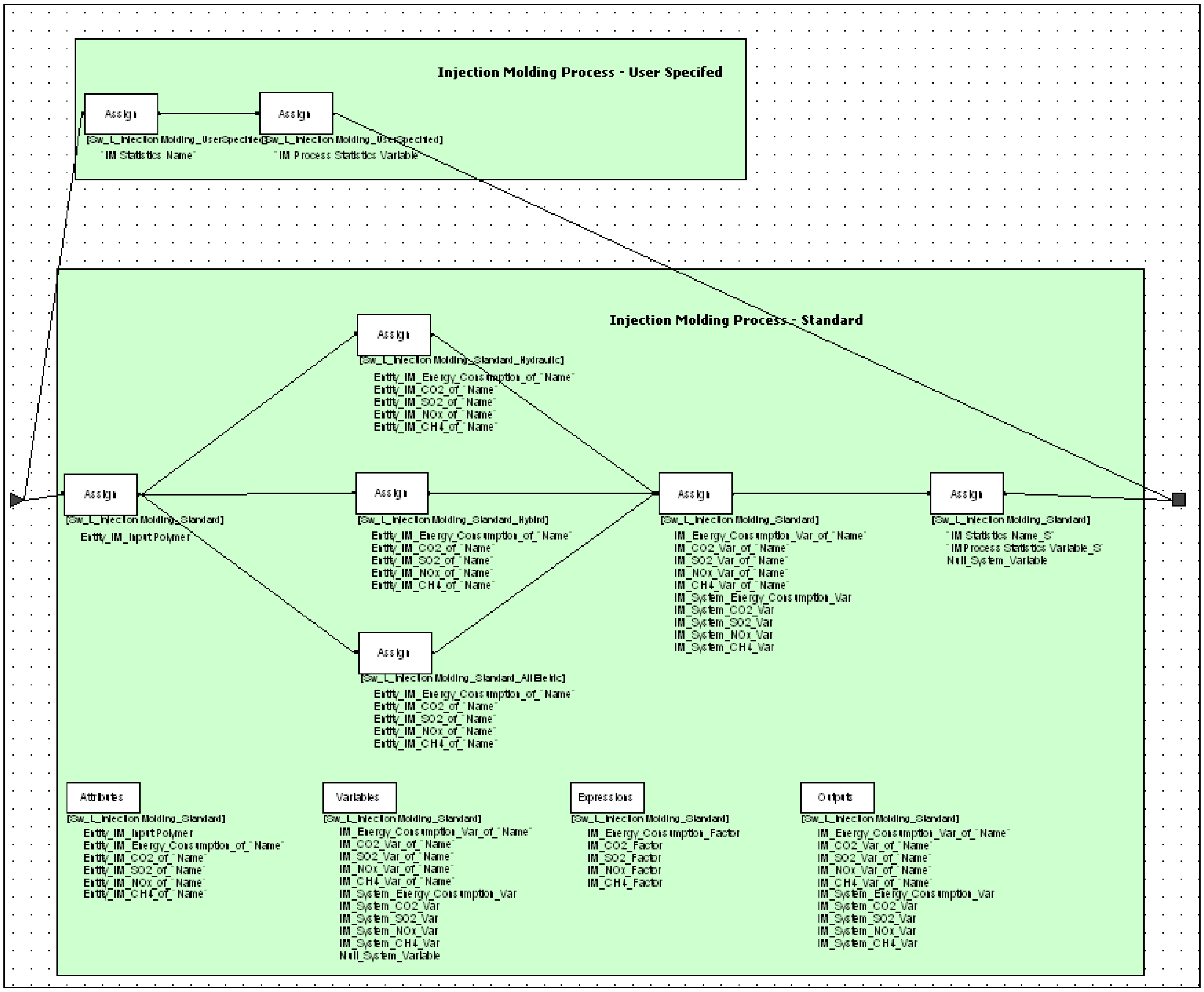


Figure B - 9: Injection Molding Process ARENA Simulation Sub-Model Logic Overview

When the entity enters the injection molding process sub-model, it first checks if this is a standard injection molding process. A user-specified injection molding process will calculate and update the user-specified measures and then the entity exits the injection molding sub-model. A standard injection molding process will first determine the injection molding machine type. Depending on the hydraulic machine, the all-electric machine and the hybrid machine used, the

energy coefficients are assigned. Pre-defined environmental performance measures are calculated and updated. After collecting the standard measures, the entity checks if there are additional user-specified statistics. The additional user-specified statistics are collected and updated. The entity eventually exits the injection molding process sub-model.

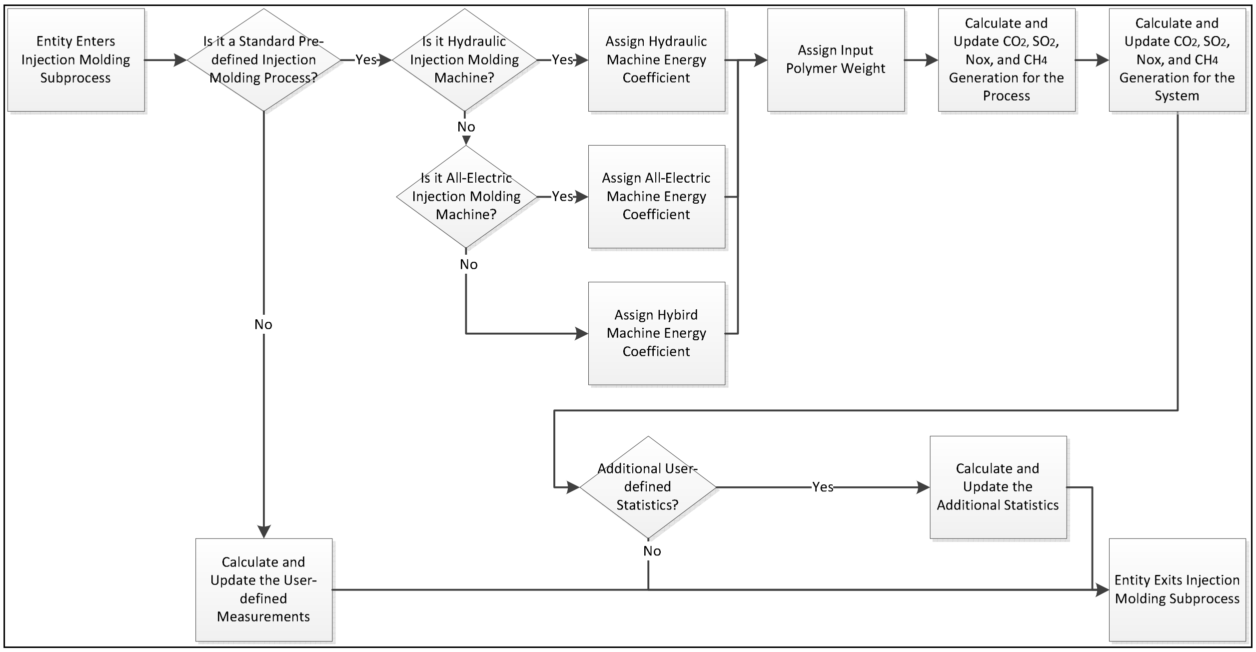


Figure B - 10: Injection Molding Process Sub-Model Flow Chart View

Plastics Processing Sub-Model

Figure B-11 shows the plastics processing ARENA simulation sub-model logic view. Figure B-12 shows the corresponding logic flow chart view for demonstration. When the entity enters the plastics processing sub-model, it first checks if this is a standard process. A user-specified plastics processing process will calculate and update the user-specified statistics and then the entity exits the sub-model. A standard coating process will first determine the plastic type. Depending on the type of plastic processed, energy coefficients are assigned. Pre-defined environmental performance measures are updated. After collecting the standard measures, the

entity checks if there are additional user-specified statistics. The additional user-specified statistics are collected and updated before the entity leaves the plastics processing process.

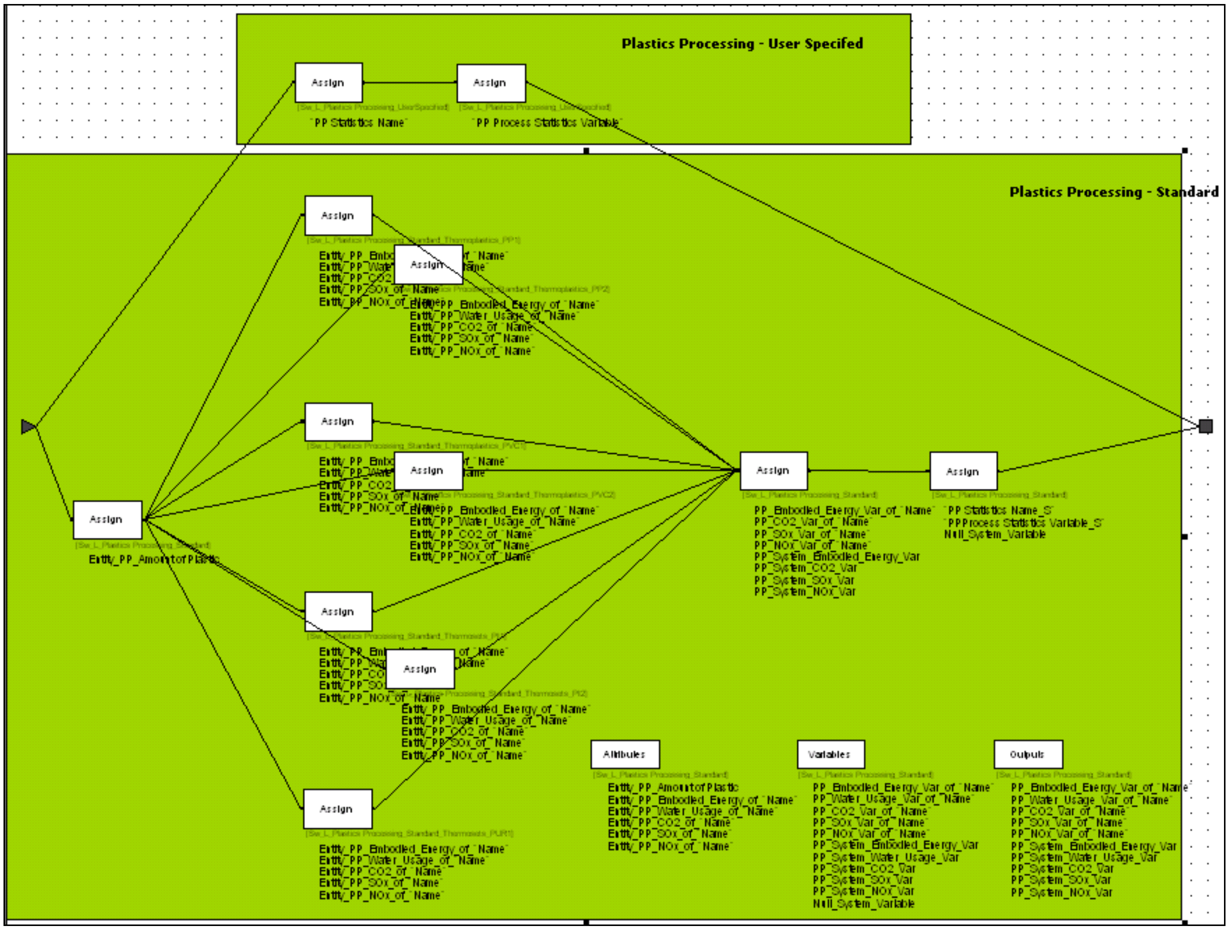


Figure B - 11: Plastics Processing ARENA Simulation Sub-Model Logic Overview

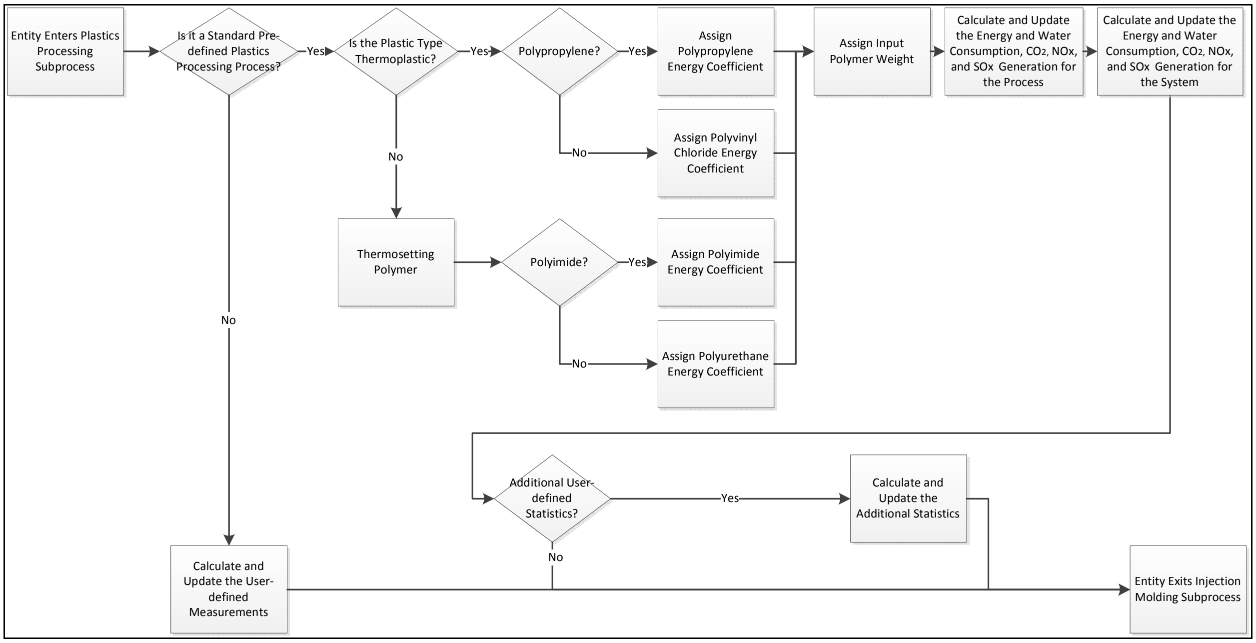


Figure B - 12: Plastics Processing Sub-Model Flow Chart View

B-3. Module Underline Logic SIMAN Code

The simulation SIMAN codes for the process with emissions module is presented in this section. The industrial coating process, the injection molding process and the plastics processing sub-model SIMAN codes are included as well.

Process with Emissions module

```
0$              DELAY:          0.0,,Other:NEXT(2$);

2$              QUEUE,          .Queue;
3$              SEIZE,          1,Other:
                                ,1:NEXT(7$);

7$              DELAY:          ToBaseTime(),,Other:NEXT(4$);

4$              RELEASE:        ,1:NEXT(9$);

5$              DELAY:          0.0,,Other:NEXT(8$);

8$              DELAY:          ToBaseTime(),,Other:NEXT(6$);

6$              DELAY:          0.0,,Other:NEXT(9$);

9$              ASSIGN;
10$             ASSIGN:         = + :NEXT(1$);

11$             ASSIGN:         Entity_Paint_Consumption_of_=Paint_Consumption_Factor(1)
* Entity_Surface_Area_of_()/ :
                                Entity_VOC_Generation_of_=VOC_Generation_Factor(1) *
```

```
Entity_Surface_Area_of_()/ :
                             Paint_Statistics_of_=Paint_Statistics_of_ +
Entity_Paint_Consumption_of_:
                             Solvent_Statistics_of_=Paint_Statistics_of_ *  / 100:
                             VOC_Statistics_of_=VOC_Statistics_of_ +
Entity_VOC_Generation_of_:
                             System_Paint_Statistics=System_Paint_Statistics +
Entity_Paint_Consumption_of_:
                             System_Solvent_Statistics=System_Solvent_Statistics +
Entity_Paint_Consumption_of_ *  / 100:
                             System_VOC_Statistics=System_VOC_Statistics +
Entity_VOC_Generation_of_;
15$           ASSIGN:        Entity_Coating Efficiency:
                             Entity_Solvent Ratio:
                             Entity_Surface Area;
13$           ASSIGN::       Null_System_Variable=0;
14$           ASSIGN:        = + :
                             Null_System_Variable=0:NEXT(1$);

12$           ASSIGN:        Entity_Paint_Consumption_of_=Paint_Consumption_Factor(2)
*  Entity_Surface_Area_of_ ()/ :
                             Entity_VOC_Generation_of_=VOC_Generation_Factor(2) *
Entity_Surface_Area_of_()/ :
                             Paint_Statistics_of_=Paint_Statistics_of_ +
Entity_Paint_Consumption_of_:
                             Solvent_Statistics_of_=Paint_Statistics_of_ *  / 100:
                             VOC_Statistics_of_=VOC_Statistics_of_ +
Entity_VOC_Generation_of_:
                             System_Paint_Statistics=System_Paint_Statistics +
Entity_Paint_Consumption_of_:
                             System_Solvent_Statistics=System_Solvent_Statistics +
Entity_Paint_Consumption_of_ *  / 100:
                             System_VOC_Statistics=System_VOC_Statistics +
Entity_VOC_Generation_of_:NEXT(15$);

1$            DELAY:         0.0,,Other;
16$           ASSIGN;
17$           ASSIGN:        = + :NEXT(1$);

18$           ASSIGN;
19$           ASSIGN:        = + :NEXT(1$);

20$           ASSIGN:        Entity_IM_Input Polymer;
21$           ASSIGN:
Entity_IM_Energy_Consumption_of_=IM_Energy_Consumption_Factor(1) * Entity_IM_Input
Polymer:
                             Entity_IM_CO2_of_=IM_CO2_Factor(1) * Entity_IM_Input
Polymer:
                             Entity_IM_SO2_of_=IM_SO2_Factor(1) * Entity_IM_Input
Polymer:
                             Entity_IM_NOx_of_=IM_NOx_Factor(1) * Entity_IM_Input
Polymer:
                             Entity_IM_CH4_of_=IM_CH4_Factor(1) * Entity_IM_Input
Polymer;
22$           ASSIGN:
IM_Energy_Consumption_Var_of_=IM_Energy_Consumption_Var_of_ +
Entity_IM_Energy_Consumption_of_:
                             IM_CO2_Var_of_=IM_CO2_Var_of_ + Entity_IM_CO2_of_:
                             IM_SO2_Var_of_=IM_SO2_Var_of_ + Entity_IM_SO2_of_:
                             IM_NOx_Var_of_=IM_NOx_Var_of_ + Entity_IM_NOx_of_:
                             IM_CH4_Var_of_=IM_CH4_Var_of_ + Entity_IM_CH4_of_:
                             IM_System_Energy_Consumption_Var=
                             IM_System_Energy_Consumption_Var +
```

```
Entity_IM_Energy_Consumption_of_:
                             IM_System_CO2_Var=IM_System_CO2_Var + Entity_IM_CO2_of_:
                             IM_System_SO2_Var=IM_System_SO2_Var + Entity_IM_SO2_of_:
                             IM_System_NOx_Var=IM_System_NOx_Var + Entity_IM_NOx_of_:
                             IM_System_CH4_Var=IM_System_CH4_Var + Entity_IM_CH4_of_;
23$           ASSIGN::       = + :
                             Null_System_Variable=0:NEXT(1$);

24$           ASSIGN:
Entity_IM_Energy_Consumption_of_=IM_Energy_Consumption_Factor(2) * Entity_IM_Input
Polymer:
                             Entity_IM_CO2_of_=IM_CO2_Factor(2) * Entity_IM_Input
Polymer:
                             Entity_IM_SO2_of_=IM_SO2_Factor(2) * Entity_IM_Input
Polymer:
                             Entity_IM_NOx_of_=IM_NOx_Factor(2) * Entity_IM_Input
Polymer:
                             Entity_IM_CH4_of_=IM_CH4_Factor(2) * Entity_IM_Input
Polymer:NEXT(22$);

25$           ASSIGN:
Entity_IM_Energy_Consumption_of_=IM_Energy_Consumption_Factor(3) * Entity_IM_Input
Polymer:
                             Entity_IM_CO2_of_=IM_CO2_Factor(3) * Entity_IM_Input
Polymer:
                             Entity_IM_SO2_of_=IM_SO2_Factor(3) * Entity_IM_Input
Polymer:
                             Entity_IM_NOx_of_=IM_NOx_Factor(3) * Entity_IM_Input
Polymer:
                             Entity_IM_CH4_of_=IM_CH4_Factor(3) * Entity_IM_Input
Polymer:NEXT(22$);

26$           ASSIGN;
27$           ASSIGN:        = + :NEXT(1$);

28$           ASSIGN::       = + :
                             Null_System_Variable=0:NEXT(1$);

29$           ASSIGN:        Entity_PP_Amount of Plastic;
30$           ASSIGN:        Entity_PP_Embodied_Energy_of_=UNIF(9340, 10300) *
Entity_PP_Amount of Plastic:
                             Entity_PP_Water_Usage_of_=0:
                             Entity_PP_CO2_of_=UNIF(5.61, 6.2) * Entity_PP_Amount of
Plastic:
                             Entity_PP_SOx_of_=UNIF(0.0498, 0.0551) * Entity_PP_Amount
of Plastic:
                             Entity_PP_NOx_of_=UNIF(0.0187, 0.0207) *
Entity_PP_Amount of Plastic;
31$           ASSIGN:        PP_Embodied_Energy_Var_of_=PP_Embodied_Energy_Var_of_ +
Entity_PP_Embodied_Energy_of_:
                             PP_CO2_Var_of_=PP_CO2_Var_of_ + Entity_PP_CO2_of_:
                             PP_SOx_Var_of_=PP_SOx_Var_of_ + Entity_PP_SOx_of_:
                             PP_NOx_Var_of_=PP_NOx_Var_of_ + Entity_PP_NOx_of_:

PP_System_Embodied_Energy_Var=PP_System_Embodied_Energy_Var +
Entity_PP_Embodied_Energy_of_:
                             PP_System_CO2_Var=PP_System_CO2_Var + Entity_PP_CO2_of_:
                             PP_System_SOx_Var=PP_System_SOx_Var + Entity_PP_SOx_of_:
                             PP_System_NOx_Var=PP_System_NOx_Var +
Entity_PP_NOx_of_:NEXT(28$);

32$           ASSIGN:        Entity_PP_Embodied_Energy_of_=UNIF(10300, 11500) *
Entity_PP_Amount of Plastic:
```

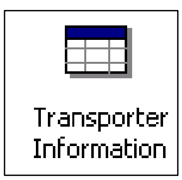
```
                             Entity_PP_Water_Usage_of_=0:
                             Entity_PP_CO2_of_=UNIF(6.46, 7.15) * Entity_PP_Amount of 
Plastic:
                             Entity_PP_SOx_of_=UNIF(0.0575, 0.0635) * Entity_PP_Amount 
of Plastic:
                             Entity_PP_NOx_of_=UNIF(0.0215, 0.0238) * 
Entity_PP_Amount of Plastic:NEXT(31$);

33$           ASSIGN:        Entity_PP_Embodied_Energy_of_=UNIF(9490, 10500) * 
Entity_PP_Amount of Plastic:
                             Entity_PP_Water_Usage_of_=0:
                             Entity_PP_CO2_of_=UNIF(5.74, 6.34) * Entity_PP_Amount of 
Plastic:
                             Entity_PP_SOx_of_=UNIF(0.051, 0.0564) * Entity_PP_Amount 
of Plastic:
                             Entity_PP_NOx_of_=UNIF(0.0191, 0.0211) * 
Entity_PP_Amount of Plastic:NEXT(31$);

34$           ASSIGN:        Entity_PP_Embodied_Energy_of_=7580 * Entity_PP_Amount of 
Plastic:
                             Entity_PP_Water_Usage_of_=UNIF(2130, 2360) * 
Entity_PP_Amount of Plastic:
                             Entity_PP_CO2_of_=UNIF(1.81, 2) * Entity_PP_Amount of 
Plastic:
                             Entity_PP_SOx_of_=UNIF(0.0192, 0.0212) * Entity_PP_Amount 
of Plastic:
                             Entity_PP_NOx_of_=UNIF(0.00638, 0.00706) * 
Entity_PP_Amount of Plastic:NEXT(31$);

35$           ASSIGN:        Entity_PP_Embodied_Energy_of_=UNIF(16100, 17900) * 
Entity_PP_Amount of Plastic:
                             Entity_PP_Water_Usage_of_=0:
                             Entity_PP_CO2_of_=UNIF(11.3, 12.5) * Entity_PP_Amount of 
Plastic:
                             Entity_PP_SOx_of_=UNIF(0.1, 0.111) * Entity_PP_Amount of 
Plastic:
                             Entity_PP_NOx_of_=UNIF(0.0376, 0.0415) * 
Entity_PP_Amount of Plastic:NEXT(31$);

36$           ASSIGN:        Entity_PP_Embodied_Energy_of_=UNIF(17800, 19600) * 
Entity_PP_Amount of Plastic:
                             Entity_PP_Water_Usage_of_=0:
                             Entity_PP_CO2_of_=UNIF(12.6, 13.9) * Entity_PP_Amount of 
Plastic:
                             Entity_PP_SOx_of_=UNIF(0.112, 0.124) * Entity_PP_Amount 
of Plastic:
                             Entity_PP_NOx_of_=UNIF(0.042, 0.0464) *  Entity_PP_Amount 
of Plastic:NEXT(31$);

37$           ASSIGN:        Entity_PP_Embodied_Energy_of_=UNIF(10900, 12100) * 
Entity_PP_Amount of Plastic:
                             Entity_PP_Water_Usage_of_=UNIF(7310, 8080) * 
Entity_PP_Amount of Plastic:
                             Entity_PP_CO2_of_=UNIF(4.57, 5.28) * Entity_PP_Amount of 
Plastic:
                             Entity_PP_SOx_of_=UNIF(0.0523, 0.0578) * Entity_PP_Amount 
of Plastic:
                             Entity_PP_NOx_of_=UNIF(0.0174, 0.0193) * 
Entity_PP_Amount of Plastic:NEXT(31$);
```

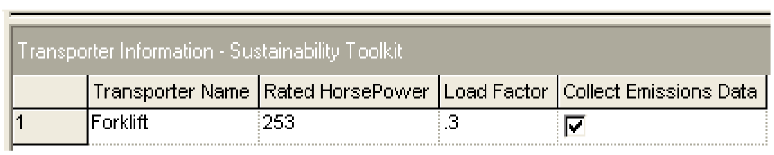
Appendix C. ARENA Template Development for Material Handling Modules

Appendix C-1 displays the material handling modules user interfaces. Appendix C-2 contains the logic flow which is carried out within the modules. Appendix C-3 shows the underline logic SIMAN codes of the modules. Details of the development and implementation of material handling modules can be found in section 5.3.

C-1. Module Dialogue Interface

Figure C-1 shows the user interface of the transporter information data module. When the user clicks on the data module, a table opens up for the user to enter the transporter information. The transporter is named in the *Transporter Name* field. The transporter's horsepower is specified in the *Rated Horsepower* field. The transporter's load factor is entered in the *Load Factor* field. The *Collect Emissions Data* field includes a checkbox. Emission measures for running the transporter will be collected and reported with the checkbox checked.





Transporter Information - Sustainability Toolkit

	Transporter Name	Rated HorsePower	Load Factor	Collect Emissions Data
1	Forklift	253	.3	☑

Figure C - 1: Transporter Information Data Module Interface

Figure C-2 depicts the request with emissions module interface. The transporter requested is selected in the *Transporter Name* field. The *Selection Rule* is used to specify the rule of selecting the requested transporter. The user can choose to save the transporter number to an entity attribute using the *Save Attribute* field. The user also needs to provide the information of *Priority, Entity Location,* and *Velocity.* The entity enters the queue that is specified with *Queue Name.* When the transporter reaches the entity, the transporter is determined to be left idling based on the *Idling Probability* specified in the request with emission module.

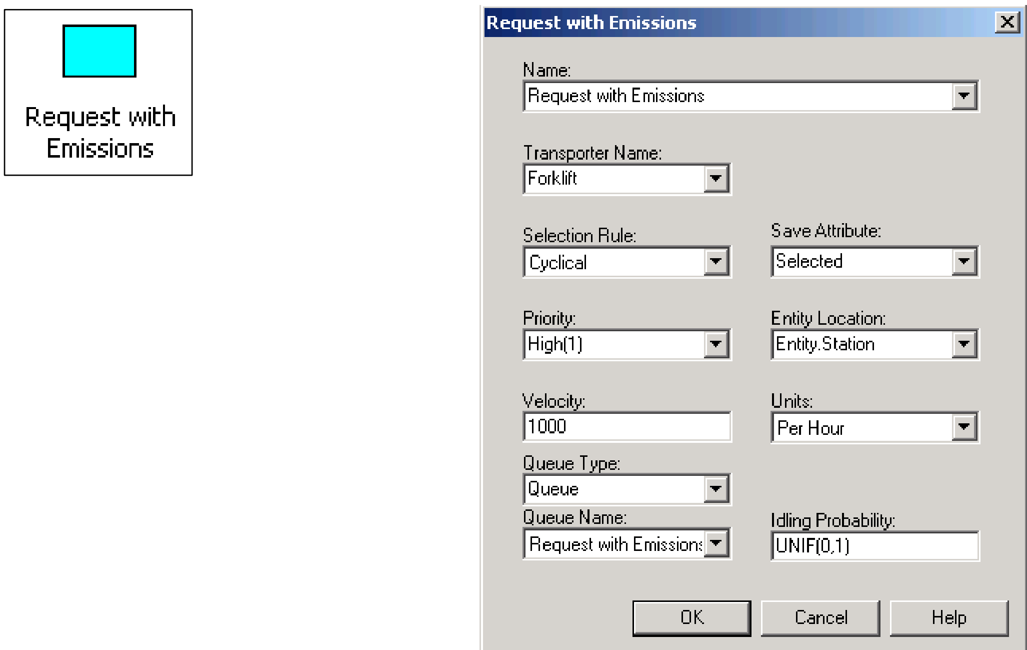


Figure C - 2: Request with Emissions Module Interface

The transport with emissions module dialogue interface is shown in Figure C-3. The *Transporter Name* and *Unite Number* fields are used to identify the transporter used. The *Entity Destination* and *Station Name* fields are used to specify the station that the transporter travels to. The user can enter the transporter *Velocity* and the *Load Factor* as well.

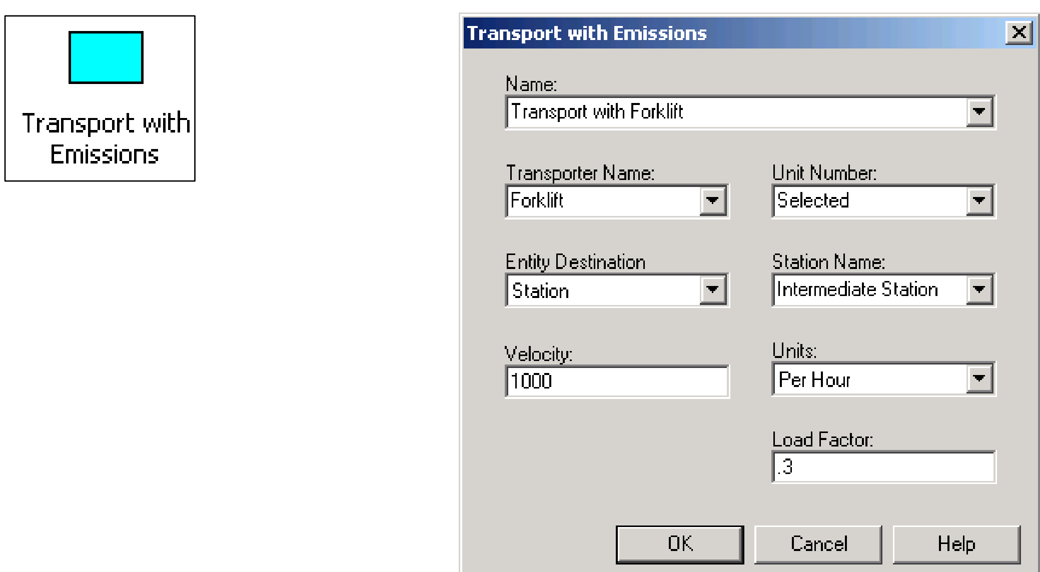
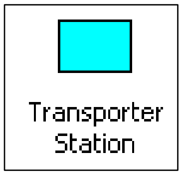


Figure C - 3: Transport with Emissions Module Interface

The station with emissions module user interface is shown in Figure C-4. The station is specified in the *Station Name* field. When the transporter arrives at the station, it will be determined to be left idling based on the probability specified in the *Idling Probability* field. The *Report Statistics* checkbox is checked to report simulation statistics for the station module.



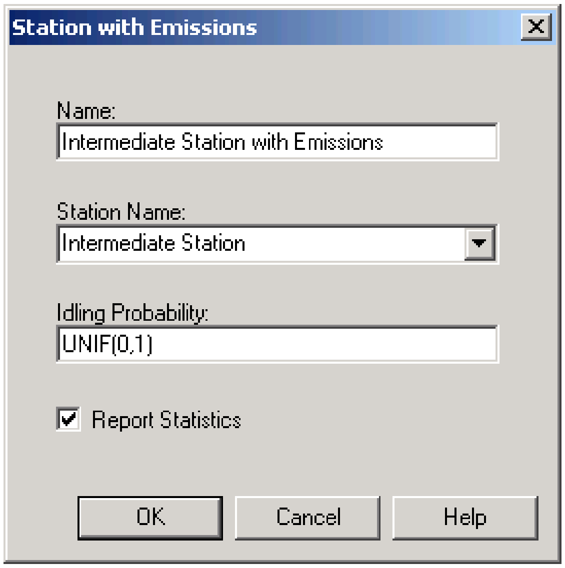
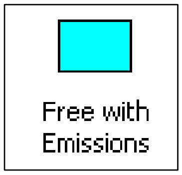


Figure C - 4: Station with Emissions Module Interface

Figure C-5 depicts the free with emissions module dialogue. The user simply needs to specify the transporter using the *Transporter Name* and *Unit Number* fields.



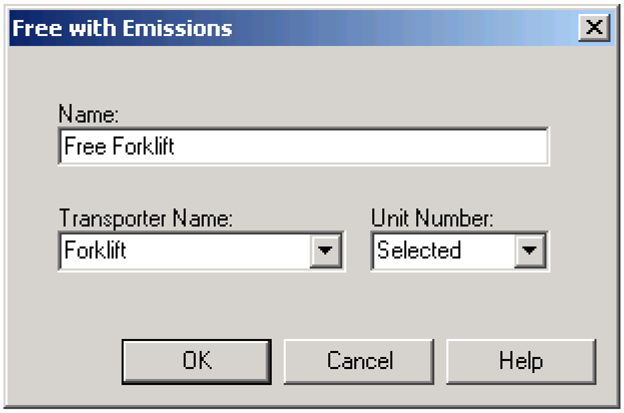


Figure C - 5: Free with Emissions Module Interface

C-2. Module Logic Window

Request with Emissions Module

Figure C-6 shows the request with emissions module logic view implemented in ARENA simulation software using standard modules. Figure C-7 shows the flow chart view of the request with emission module logic. When the entity enters the request with emissions module, it enters a queue first. If the entity is not the first in the queue, it will enter the end of the queue and wait until it becomes the first. As soon as the entity becomes the first in the queue, it allocates the requesting transporter. The transporter will be assigned for a transporter moving start time. Then the transporter is moved from its current location to the entity's location. When the transporter arrives at the entity's location, the emission measures of HC, CO, NO_x and PM are collected and updated. The transporter is determined to be left idling based on a user-specified probability. An attribute of truck idling will be assigned to "yes" if the truck is left idling and "no" if it is turned off. A starting time for loading the entity is assigned. Then the entity leaves the request with emissions module.

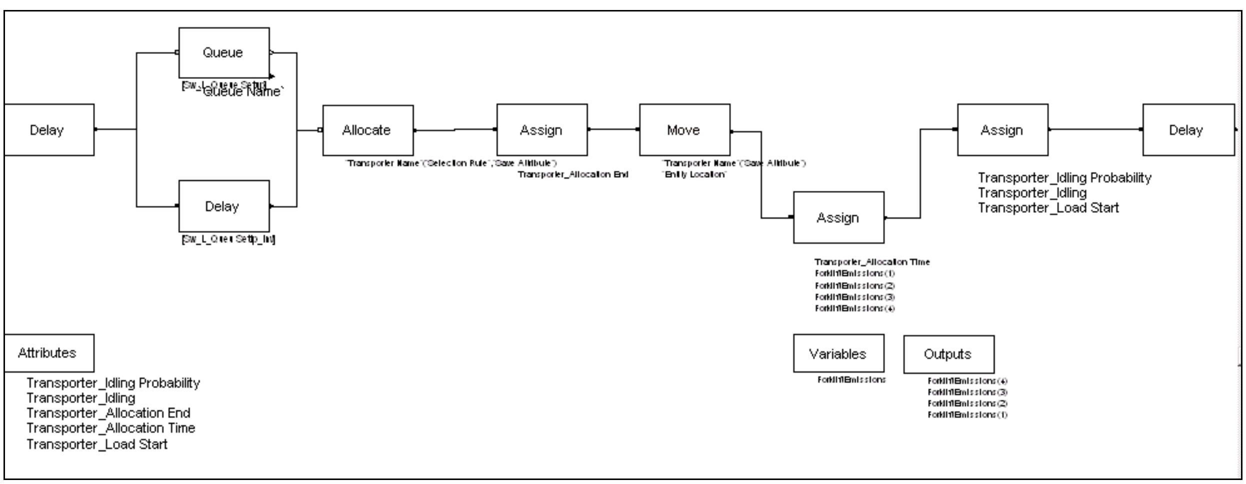


Figure C - 6: Request with Emissions Module ARENA Simulation Logic Overview

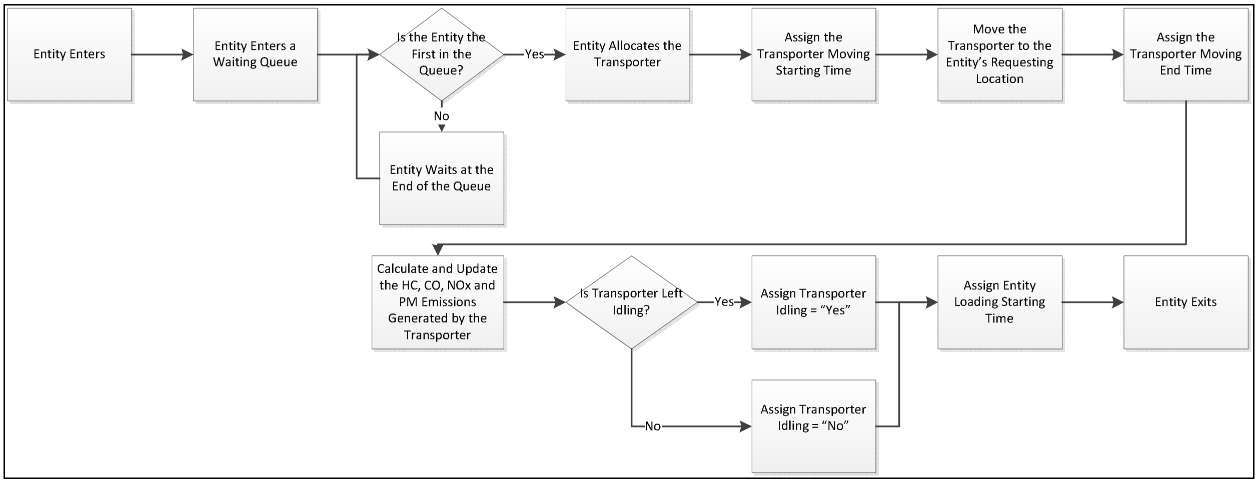


Figure C - 7: Request with Emissions Module Flow Chart View

Transport with Emissions Module

Figure C-8 depicts the transport with emissions module simulation logic overview modeled using ARENA standard modules. A flow chart view representing the simulation logic of the transport with emissions is shown in Figure C-9.

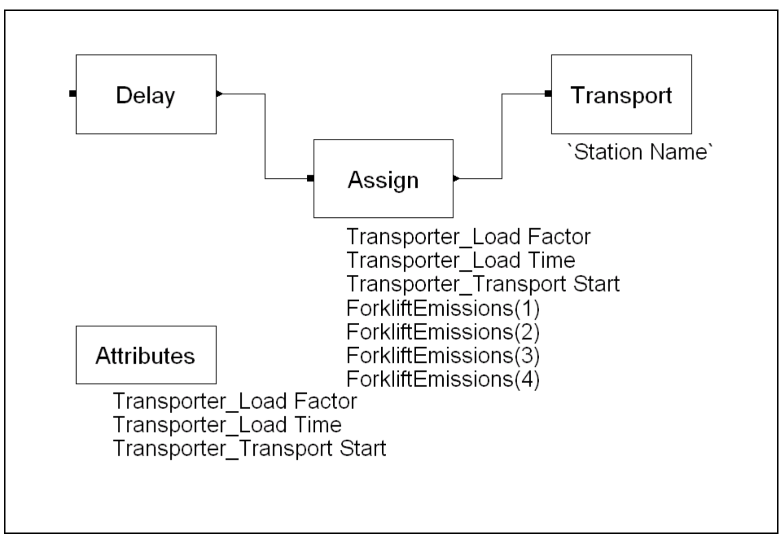


Figure C - 8: Transport with Emissions Module ARENA Simulation Logic Overview

When the entity enters the transport with emissions module, it first assigns the current simulation time as the entity loading end time. Then the entity checks the value of the transporter idling attribute value. If the value is "yes", the emission measures are calculated and updated

based on the loading time. If the value is "no", the entity skips the calculation step. The transport starting time is assigned and the entity leaves the transport with emissions module.

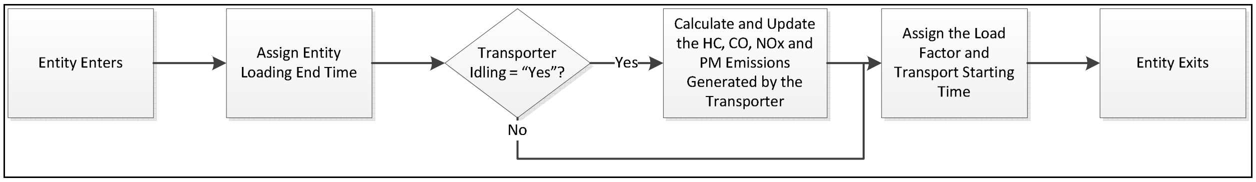


Figure C - 9: Transport with Emissions Module Flow Chart View

Transporter Station Module

The transporter station module ARENA simulation logic is shown in Figure C-10. The simulation logic is represented by flow chart view shown in Figure C-11.

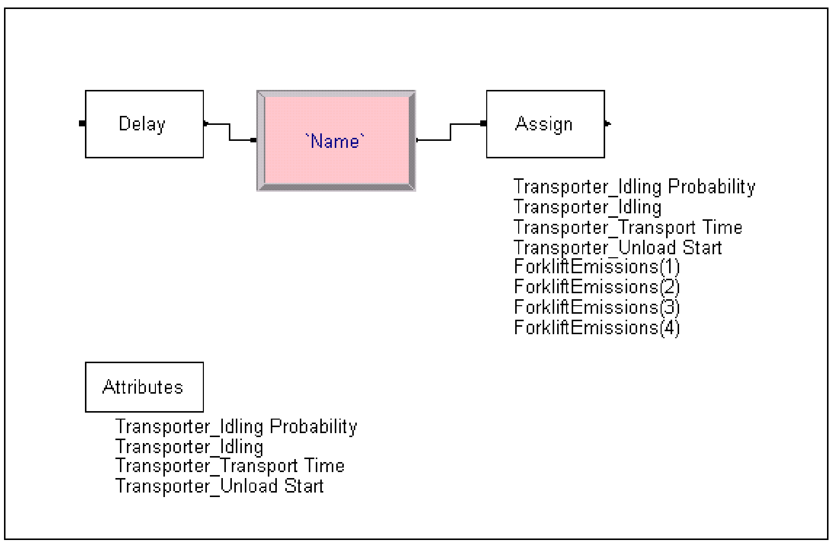


Figure C - 10: Transporter Station Module ARENA Simulation Logic Overview

When the transporter arrives at a station with the entity, the transport ending time is assigned in order to be used in calculating the emission measures. The transporter will be assigned with the transporter idling attribute ("yes" if the transporter is left idling and "no" if it is turned off). The amount of emission generated when the transporter travels to the station is calculated and updated based on the transporting time. Entity unloading starting time is assigned

to be the current simulation time. Entity remains at the station and starts to be unloaded from the transporter.

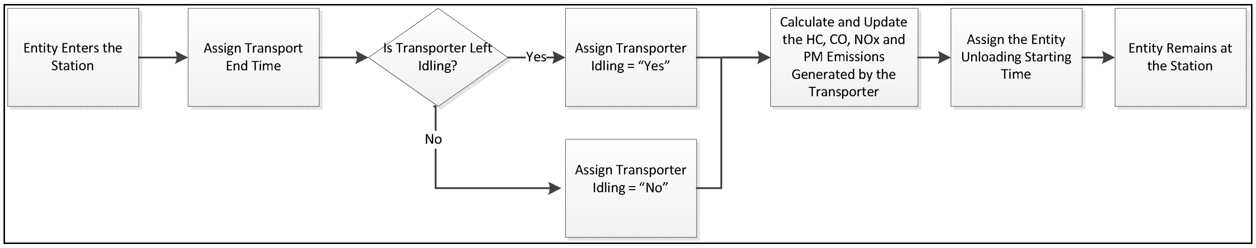


Figure C - 11: Transport Station Module Flow Chart View

Free with Emissions Module

The free with emissions module simulation logic overview is shown in Figure C-12 using ARENA standard modeling modules. To demonstrate the logic sequence, a flow chart view of the free with emissions module is presented in Figure C-13.

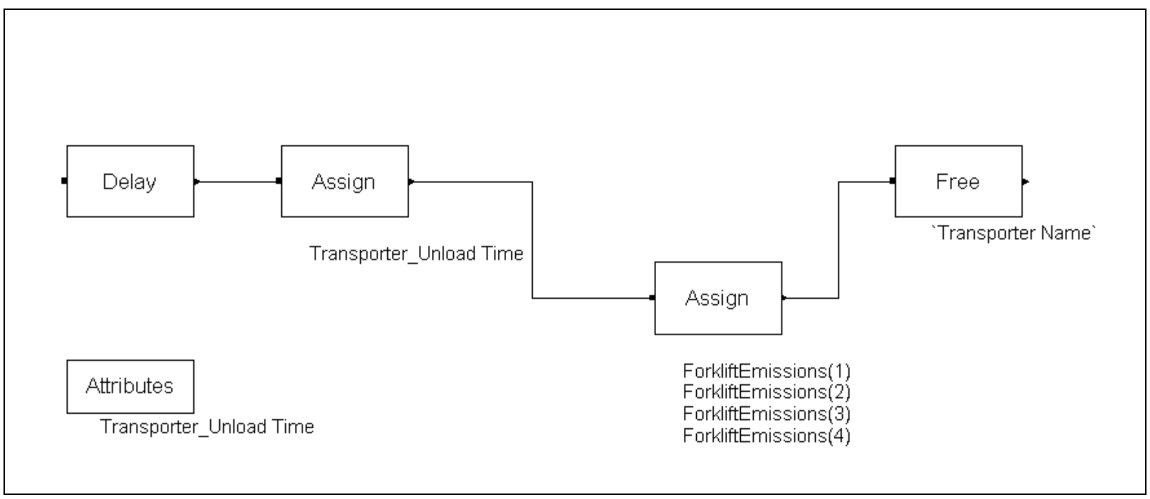


Figure C - 12: Free with Emissions Module ARENA Simulation Logic Overview

When entity enters the free with emissions module, the unloading end time is assigned with the value of the current simulation time. The entity evaluates the transporter idling attribute value. If the value is "yes" meaning the transporter is idling during the unloading time, the amount of emission (HC, CO, NO_x and PM) generated is calculate and the measures are

updated. If the idling value is “no”, the calculation step will be skipped. The entity frees the transporter and leaves the free with emissions module.

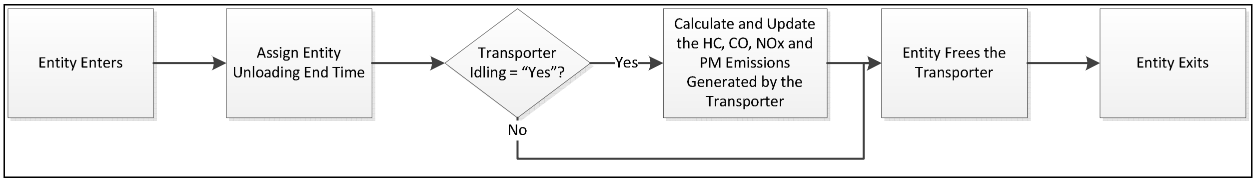


Figure C - 13: Free with Emissions Module Flow Chart View

C-3. Module Underline Logic SIMAN Code

The following section lists the simulation SIMAN codes for the request with emission module, the transport with emissions module, the transporter station module and the free with emissions module.

Request with Emissions

```
5$            DELAY:         0.0,,Other:NEXT(3$);

3$            QUEUE;
1$            ALLOCATE,      1:(,);
2$            ASSIGN:        Transporter_Allocation End=TNOW;
0$            MOVE:          (),,/ToBaseTime(1);
6$            ASSIGN:        Transporter_Allocation Time=TNOW - Transporter_Allocation
End:
                             ForkliftEmissions(1)=
                             ForkliftEmissions(1)+ (Load Factor*Rated
HorsePower*EmissionFactor(1)*Transporter_Allocation Time):
                             ForkliftEmissions(2)=
                             ForkliftEmissions(2)+ (Load Factor*Rated
HorsePower*EmissionFactor(2)*Transporter_Allocation Time):
                             ForkliftEmissions(3)=
                             ForkliftEmissions(3)+ (Load Factor*Rated
HorsePower*EmissionFactor(3)*Transporter_Allocation Time):
                             ForkliftEmissions(4)=
                             ForkliftEmissions(4)+ (Load Factor*Rated
HorsePower*EmissionFactor(4)*Transporter_Allocation Time);
7$            ASSIGN:        Transporter_Idling Probability:
                             Transporter_Idling=DISC(Transporter_Idling Probability,
1, 1.0, 0):
                             Transporter_Load Start=TNOW;
8$            DELAY:         0.0,,Other;
4$            DELAY:         0.0,,Other:NEXT(1$);
```

Transport with Emissions

```
0$            DELAY:         0.0,,Other:NEXT(1$);

1$            ASSIGN:        Transporter_Load Factor:
```

```
                              Transporter_Load Time=TNOW - Transporter_Load Start:
                              Transporter_Transport Start=TNOW:
                              ForkliftEmissions(1)=
                              ForkliftEmissions(1)+ (Transporter_Idling*Load
Factor*Rated HorsePower*EmissionFactor(1)*Transporter_Load Time):
                              ForkliftEmissions(2)=
                              ForkliftEmissions(2)+ (Transporter_Idling*Load
Factor*Rated HorsePower*EmissionFactor(2)*Transporter_Load Time):
                              ForkliftEmissions(3)=
                              ForkliftEmissions(3)+ (Transporter_Idling*Load
Factor*Rated HorsePower*EmissionFactor(3)*Transporter_Load Time):
                              ForkliftEmissions(4)=
                              ForkliftEmissions(4)+ (Transporter_Idling*Load
Factor*Rated HorsePower*EmissionFactor(4)*Transporter_Load Time);
2$             TRANSPORT:     ,,/ToBaseTime(1);
```

Transporter Station

```
2$             DELAY:         0.0,,Other:NEXT(0$);
0$             STATION,       Station 1;
5$             DELAY:         0.0,,VA:NEXT(1$);

1$             ASSIGN:        Transporter_Idling Probability:
                              Transporter_Idling=DISC(Transporter_Idling Probability,
1, 1.0, 0):
                              Transporter_Transport Time=TNOW - Transporter_Transport
Start:
                              Transporter_Unload Start=TNOW:
                              ForkliftEmissions(1)=
                              ForkliftEmissions(1)+ (Transporter_Load Factor*Rated
HorsePower*EmissionFactor(1)*Transporter_Transport Time):
                              ForkliftEmissions(2)=
                              ForkliftEmissions(2)+ (Transporter_Load Factor*Rated
HorsePower*EmissionFactor(2)*Transporter_Transport Time):
                              ForkliftEmissions(3)=
                              ForkliftEmissions(3)+ (Transporter_Load Factor*Rated
HorsePower*EmissionFactor(3)*Transporter_Transport Time):
                              ForkliftEmissions(4)=
                              ForkliftEmissions(4)+ (Transporter_Load Factor*Rated
HorsePower*EmissionFactor(4)*Transporter_Transport Time);
```

Free with Emissions

```
0$             DELAY:         0.0,,Other:NEXT(1$);

1$             ASSIGN:        Transporter_Unload Time=TNOW - Transporter_Unload Start;
2$             ASSIGN:        ForkliftEmissions(1)=
                              ForkliftEmissions(1)+ (Transporter_Idling*Load
Factor*Rated HorsePower*EmissionFactor(1)*Transporter_Unload Time):
                              ForkliftEmissions(2)=
                              ForkliftEmissions(2)+ (Transporter_Idling*Load
Factor*Rated HorsePower*EmissionFactor(2)*Transporter_Unload Time):
                              ForkliftEmissions(3)=
                              ForkliftEmissions(3)+ (Transporter_Idling*Load
Factor*Rated HorsePower*EmissionFactor(3)*Transporter_Unload Time):
                              ForkliftEmissions(4)=
                              ForkliftEmissions(4)+ (Transporter_Idling*Load
Factor*Rated HorsePower*EmissionFactor(4)*Transporter_Unload Time);
3$             FREE;
```

Appendix D. ARENA Template Development for General Collection Modules

Appendix D-1 displays the general collection modules user interfaces. Appendix D-2 contains the logic flow which is carried out within the general modules. Appendix D-3 shows the underline logic SIMAN codes of the modules. Details of the development and implementation of general collection modules can be found in section 5.4.

D-1. Module Dialogue Interface

Figure D-1 shows the user interface of the general collection data module. When the user clicks on the data module, a table opens up where the user can define a collection. The user can name the collection in the *Collection Name* field. Entity accumulator or system accumulator is specified in the *Collection Type* field. The *Rows* and *Member Value* fields are used for entity accumulators only, and define the entity array and index. The *Accumulator Expression* field is used to define the collection evaluation formula. If a difference value (delta) is used in the *Accumulator Expression*, the *Delta* checkbox needs to be checked and delta type is selected in the *Delta Type* field.

Collection - Sustainability Toolkit

	Collection Name	Collection Type	Rows	Member Value	Delta	Delta Type	Accumulator Expression
1	CO2 generation	Entity Accumulator	4	4 rows	☑	TNOW	Delta * Truck HP * Truck Ideling * EmissionCoe(1)
2	NOx generation	Entity Accumulator	4	4 rows	☑	TNOW	Delta * Truck HP * Truck Ideling * EmissionCoe(2)
3	System CO2 generation	System Accumulator		0 rows	☐	TNOW	CO2 generated by truck
4	System NOx generation	System Accumulator		0 rows	☐	TNOW	NOx generated by truck

Figure D - 1: Collection Data Module Interface

Figure D-2 shows the user modeling interface for begin collection module. The user simply needs to choose the collection type (entity accumulator or system accumulator) form the *Type* drop-down menu. The specific collection is then chosen from the *Collection Name* drop-down menu.

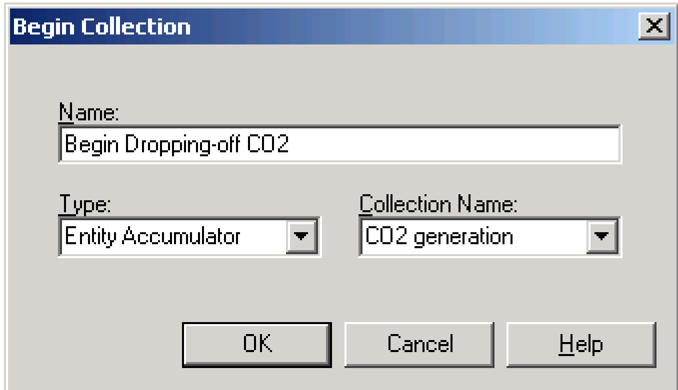


Figure D - 2: Begin Collection Module Interface

The end collection module user interface is shown in Figure D-3. The *Type* and *Collection Name* fields are used to choose a pre-defined collection to end. For an entity accumulator, an *Define Member Value* field will appear with two additional drop-down menus of *Define Memember Value by* and *Member Value.* As shown in Figure D-1, these two drop-down menus define how to evaluate the entity array specified in this entity accumlator. The entity accumulator and system accumulator both have a *Saving Attribute* field to save the evaluated accumulator value to an entity attribute.

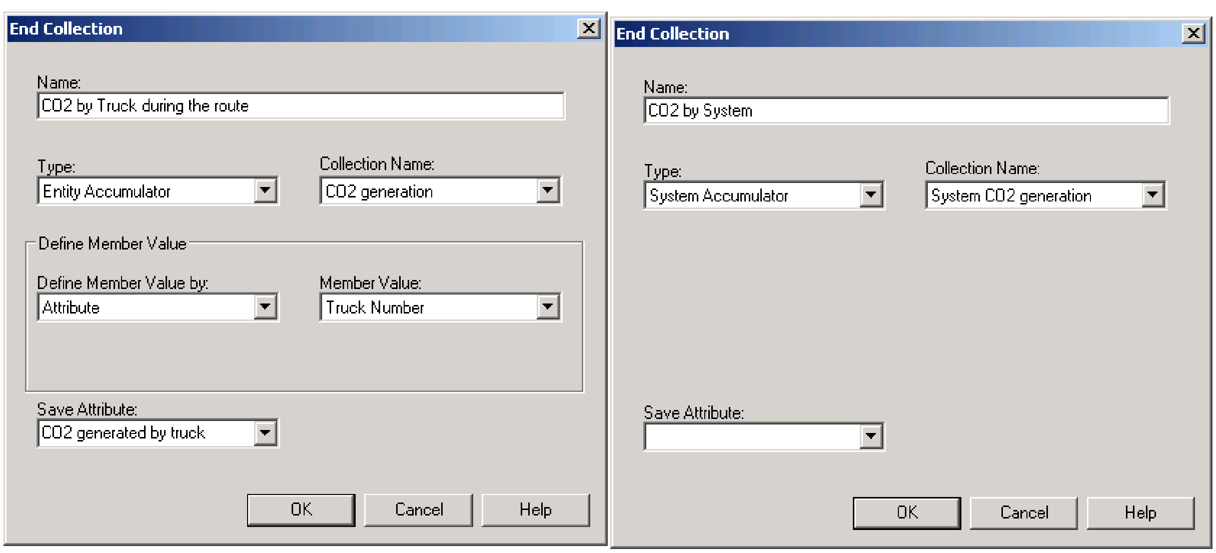


Figure D - 3: End Collection Module Interface

D-2. Module Logic Window

Begin Collection module

Figure D-4 shows the begin collection module logic overview implemented in ARENA simulation software. Standard ARENA modeling modules are used to constrate the logic. To demenstrate the simulation logic sequence, Figure D-5 shows the flow chart view of the begin collection module. When the entity enters the begin collection module, a collection, which is defined in the collection data module, is specified. A collection pairing checking attribute is assigned to have the value of "Yes". The begin collection moduel then determines whether the pre-defined collection include delta in its evaluation expression. If no delta is used, the entity leaves the begin collection module. If delta is used in the accumulator expression, the module will assign an initial value of the delta. A TNOW type of delta has an initial value of the current simulation time. An attribute or expression type of delta has an intiail value of the current value of the corresonding attribute or expression. The entity leaves the begin collection module after assigning the intial value for the delta.

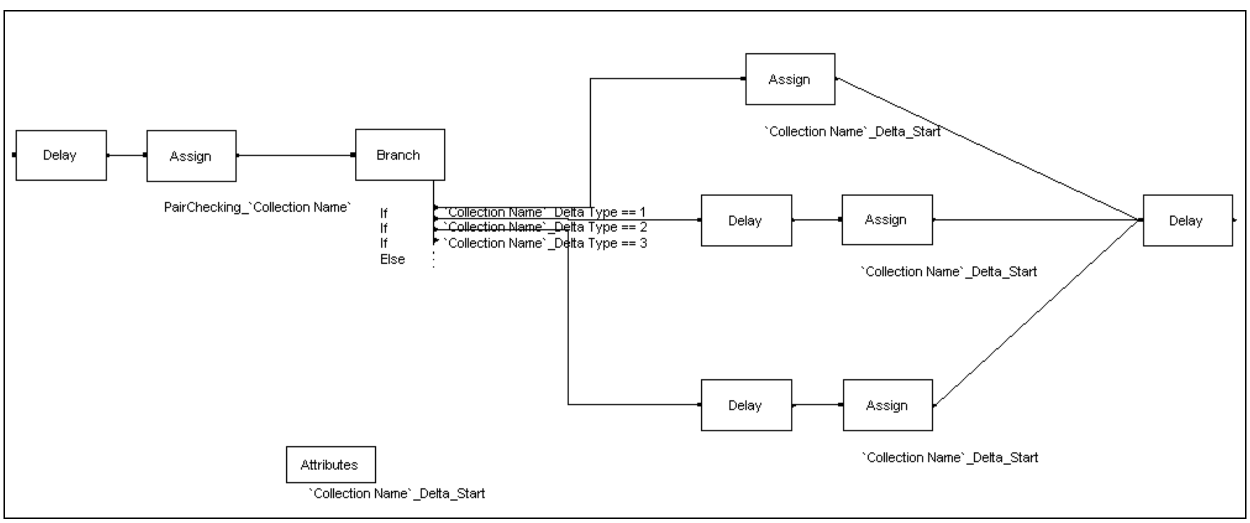


Figure D - 4: Begin Collection Module ARENA Simulation Logic Overview

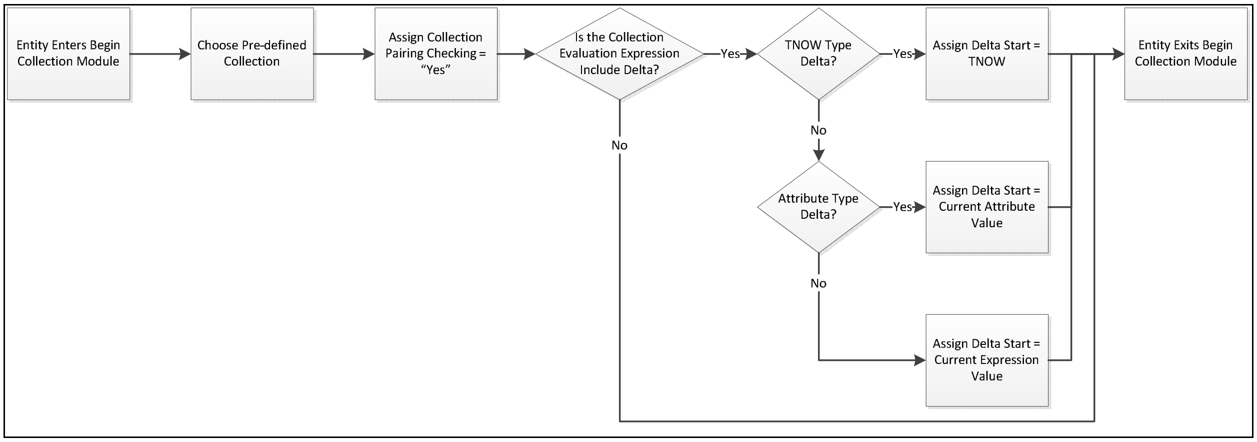


Figure D - 5: Begin Collection Module Flow Chart View

End Collection module

The end collection module ARENA simulation logic overview is shown in Figure D-6. The demonstrating flow chart view of the end collection module is shown in Figure D-7. When the entity enters the end collection module, a collection, which is defined in the collection data module, is specified. Then the module checks for the collection pairing checking attribute value. If the paring attribute value is not "yes", a warning dialogue will be displayed indicating that the specified end collection dose not have a coressponding begin collection in the model. If the paring attribute value is "yes", the end collection moduel then determines whether the pre-defined collection include delta in its evaluation expression. If the delta is used in the accumulator expression, the module will assign an ending value of the delta and evaluate the delta final value. If no delta is used, the module skips the step for calculating the delta value. Based on the accumulator type, the end collection module calculates and updates the defined collection using the accumulator expression.

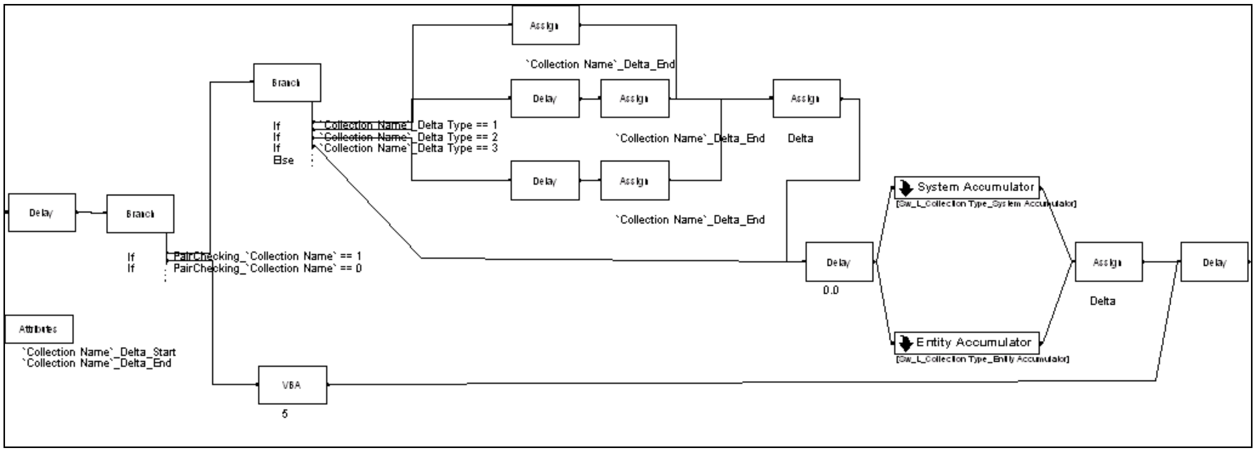


Figure D - 6: End Collection Module ARENA Simulation Logic Overview

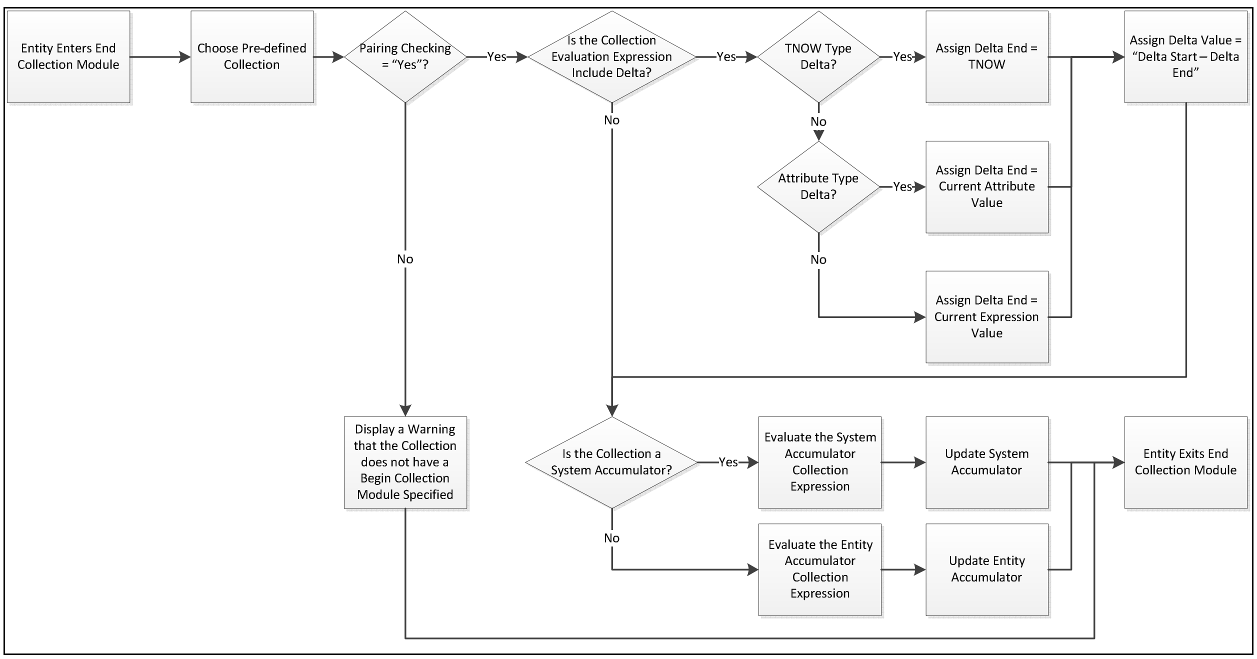


Figure D - 7: End Collection Module Flow Chart View

D-3. Module Underline Logic SIMAN Code

The simulation SIMAN codes for the begin collection module and the end collection module are presented in this section.

Begin Collection module

```
0$              DELAY:          0.0,,Other:NEXT(2$);

2$              ASSIGN:         PairChecking_=1;
3$              BRANCH,         1:
                                If,_Delta Type == 1,4$,No:
                                If,_Delta Type == 2,7$,No:
                                If,_Delta Type == 3,8$,No:
                                Else,,No;
4$              ASSIGN:         _Delta_Start=TNOW;
1$              DELAY:          0.0,,Other;
7$              DELAY:          0.0,,Other:NEXT(5$);

5$              ASSIGN:         _Delta_Start=_Delta Expression:NEXT(1$);

8$              DELAY:          0.0,,Other:NEXT(6$);

6$              ASSIGN:         _Delta_Start=_Delta Expression:NEXT(1$);
```

End Collection module

```
1$              DELAY:          0.0,,Other:NEXT(4$);

4$              BRANCH,         1:
                                If,PairChecking_ == 1,6$,No:
                                If,PairChecking_ == 0,5$,No;
6$              BRANCH,         1:
                                If,_Delta Type == 1,7$,No:
                                If,_Delta Type == 2,10$,No:
                                If,_Delta Type == 3,11$,No:
                                Else,12$,No;
7$              ASSIGN:         _Delta_End=TNOW;
2$              ASSIGN:         Delta=_Delta_End - _Delta_Start;
12$             DELAY:          0.0,,Other:NEXT(13$);

13$             ASSIGN:         SystemAccum_=SystemAccum_ + ( SystemAccum__Exp );
14$             ASSIGN:         =SystemAccum__Exp:NEXT(3$);

15$             DELAY:          0.0,,Other:NEXT(3$);

3$              ASSIGN:         Delta=0;
0$              DELAY:          0.0,,Other;
10$             DELAY:          0.0,,Other:NEXT(8$);

8$              ASSIGN:         _Delta_End=_Delta Expression:NEXT(2$);

11$             DELAY:          0.0,,Other:NEXT(9$);

9$              ASSIGN:         _Delta_End=_Delta Expression:NEXT(2$);

5$              VBA:            1,vba:NEXT(0$);

19$             BRANCH,         1:
                                If, == EntityAccum__ValuesCollected (EntityAccum
Indexing),16$,No:
```

```
                              If,
                               <> EntityAccum__ValuesCollected (EntityAccum Indexing)
&&  EntityAccum Indexing < 2 * EntityAccum__Rows - 1,
                              18$,No:
                              Else,22$,No;
16$           ASSIGN:         EntityAccum_(EntityAccum Storage Indexing)=
                              EntityAccum_(EntityAccum Storage Indexing) +
( EntityAccum__Exp );
20$           ASSIGN:         =EntityAccum__Exp;
17$           ASSIGN:         EntityAccum Indexing=1:
                              EntityAccum Storage Indexing=1:NEXT(3$);

18$           ASSIGN:         EntityAccum Indexing=EntityAccum Indexing + 2:
                              EntityAccum Storage Indexing=EntityAccum Storage Indexing
+ 1:NEXT(19$);

22$           VBA:            2,vba:NEXT(17$);

21$           DELAY:          0.0,,Other:NEXT(17$);
```