Modularity in System Dynamics: representing closed-loop supply chain configurations

Aly Elmasry
European Masters programme in System Dynamics
alyelmasry0@gmail.com

Andreas Größler
Department of Operations Management, University of Stuttgart
andreas.groessler@bwi.uni-stuttgart.de

Abstract

Modularity has been used in several disciplines as a structured way to approach complexity, to increase efficiency, to reduce repetitive work, to decrease errors, and to create cumulative learning. The system dynamics field has yet to benefit from modularity as a systematic method for developing models. In this article, we discuss system dynamics modules that can be used in developing supply chain models. First, certain design rules are identified for developing modules in system dynamics. Then, a theoretical framework is used to guide closed-loop supply chain module development. Afterwards, reoccurring structures are identified to be used in developing system dynamics modules for modelling supply chain systems. Finally, the modules were used to represent three different closed-loop supply chain configurations; we show the implications of each configuration.

Introduction

Developing system dynamics models is an extensive process that involves identifying problem variables, developing system structure, and analysing dynamic behaviour, along with validation tasks on each of these steps (Sterman, 2000: Chapter 3). Modularity simplifies this process, by providing ready-made modules that could be plugged together to represent the system under study. Although there have been previous attempts to develop generic model building blocks for model development, for example the system dynamics molecules (Hines, 1996), there are no attempts to apply modularity systematically. In addition, no application specific modules were developed in the system dynamics field, for instance for supply chain applications.

Most of system dynamics models in supply chain are developed using similar structures, for example: material flows, stock control heuristics, incoming orders processing, costs and revenue, and information flows, all of which could be represented by generic modules. This motivated us to apply modularity to supply chain applications; as Wolstenholme and Coyle (1983: 2) put it:

> "The modular approach arises from the recognition that systems in a wide variety of fields contain structurally similar elements…"

To apply modularity in supply chain modelling, we use Wikner and Tang's (2008) theoretical framework for closed-loop supply chain configurations. In addition, we review system dynamics models in supply chain management, to deduce reoccurring structures and use them in developing system dynamics modules for supply chain applications. Upon developing supply chain modules in system dynamics, we perform several validation tests to eliminate the possibility of modelling errors and increase confidence in the modules' structures and their behaviour (Forrester & Senge, 1980). Lastly, we use these modules to represent three different closed loop supply chain configurations and show the implications of each configuration.

The benefits of modularity

Modularization as a practical concept can be credited to the computer industry (Baldwin & Clark, 1997) as a way to break up computer programs into separate components. The components (i.e. modules) follow certain design rules and standards that allow them to be coupled via predefined interfaces. Baldwin and Clark (2000) argue that the power of modularity lies in the structurally successful partition of a system into blocks or modules that can be developed independently and in parallel, and then integrated together to form the whole.

Nowadays, the high promise of modularity is recognized across many different fields. Modular-construction is used today to apply standardization and predesign leading to mass production, significant reduction in on-site construction, process efficiency, and the ability to repeat orders for clients (Gibb, 2001). In the computer industry, modularization allowed for independent experimentation and design of components, leading to innovation and high rate of advancement (Baldwin & Clark, 1997). In software development, modularization allowed for ease of work distribution among software developers; fewer errors while developing software; ease of error recognition; work structuring; ease of code reading; and the ability to work in parallel (Mall, 2009).

Modularity can benefit system dynamics modelling practice in many ways. Model construction is easier and more efficient when done using previously developed modules. The reuse of well-validated and tested modules decreases modelling time, errors and costs. The modelling time decreases by dealing with complexity structurally through a ‘divide and conquer’ strategy. This allows the modeller to think systematically and to structure the modelling process. Testing modules individually could then identify errors in models. When errors are identified, and fixed, the modules could be used repeatedly in different models: helping the modeller in constructing future models faster. Furthermore, the modelling software could support class libraries, so that a structural change in a module class automatically changes all the corresponding modules in the constructed model. This will further decrease time and effort in model development and simplify adaptation.

Modules also help throughout the model validation process, in that each module would have been tested and used several times in distinct models to represent a variety of systems. When modules are used to represent multiple systems they would pass the ‘family-member’ test, which is used to “show that the model takes on the characteristics of different members of the class” where a class is a family of systems (Forrester & Senge, 1980: 25).

Modularity enhances cumulative learning by improving and building on modules. As such, modellers can share their developed modules, and others could use and improve on them. Moreover, modularity allows different modellers to work in parallel when representing one complex system. Furthermore, modellers develop an intuitive understanding of the modules’ functions and structures after using them repeatedly. This allows the modellers to see a complex system represented in modules and identify each module’s function while understanding the overall dynamics of the model. Thus, modularity simplifies complex models into multiple distinct modules, and makes it easier for the modeller to represent the system, and understand it.

Obviously, these benefits come with some limitations. First, there is the potential danger that the ease of modelling that comes with modularity can create false confidence in some of the models. Even though modules should be validated and tested thoroughly, the overall model has to be validated again structurally and behaviourally. Second, the use of modules may prevent the modeller from seeing the endogenous dynamics of the bigger system. If the model is represented in modules, the main loops in the overall system could be hidden between the modules. This can be misleading as endogenous dynamics are one of the most important foundations in system dynamics modelling (Richardson, 2011). However, modules interact with each other and if one follows the rigorous system dynamics validation practices (Forrester & Senge, 1980), these endogenous dynamics should be present in the bigger system. Although representing a complex system in modules is easier to understand, the modeller nevertheless needs to present additional representations of the system that show the main dynamics (e.g. causal loops or stock and flow diagram representing the main loops). Finally, modules may prevent the deeper understanding that comes with one's model development journey. Thus, one should study the modules thoroughly before using them.

Supply chain modules

System dynamics has a successful history of modelling forward supply chains starting from Forrester's (1961) seminal book "*Industrial dynamics*". Angerhofer and Angelides (2000) categorized system dynamics work in supply chains into three main lines of research: theory building, problem solving and improving the system dynamics approach. Interestingly, most of the models discussed use similar components to model supply chain systems, for example: ordering policy, capacity expansion, production rate, backlogging orders, product shipment and acquiring rate. Obviously, the models are structurally different to represent the system at hand. A lot of work is put to develop these models, and most of them are developed from scratch. Repeated structures are, often, presented by citing previous work. For example, Georgiadis and Besiou (2008: 1669) cite Sterman (1989) for their production ordering rate structure; and Kamath and Roy (2007) cite Forrester (1968) for similarities in model structure. Even if structures within a model are not explicitly cited from previous work, they often retain significant similarities with previously developed models. This shows that supply chain modules can be advantageous; when repeating structures are put in modules, modellers could use the modules frequently while saving considerable time and effort.

Theoretical framework and modules formulation

Complex systems consist of many elements. Elements' interconnections and interplay form the complexity and holism of the system (Baldwin & Clark, 2000). The central idea to modularity according to Baldwin and Clark (2000) is to have system elements depend on pre-defined design rules rather than on their interconnections and interplay. This allows for independent design of modules. This leads to the following design guideline:

> *Design guideline 1: Modules should depend on pre-defined design rules.*

Many design rules could be set depending upon modules' applications, functionality, and usage. As a result, design rules should be specified to guide module design and to achieve modular usefulness while retaining the module's applicability in the corresponding field and application.

Foster (1995) provides a list of recommendations that deem useful as a framework in modular design in parallel programming. Since software development is closely related to modelling and simulation, we have extracted the following design guidelines as a framework for modular design:

> *Design guideline 2: Each module should have a defined purpose with no replication of purposes among modules.*

> *Design guideline 3: Interface between modules should be clearly defined*

> *Design guideline 4: Modules should be hierarchical, and build on each other.*

Design guidelines 1-4 provide the general framework for designing system dynamics modules for supply chains. The *design guidelines* will ensure that each module could be developed with little knowledge about other modules, and allow for intuitive and easy assembly of modules to fit the purpose of modelling.

To divide the supply chain into interchangeable modules, one has to closely inspect the structure of the supply chain in system dynamics models. At first glance, it is easy to recognise that the supply chain could be divided into distinct facilities and actors: manufacturing centres, distributers, retailers, customers, acquisition centres, and recovery centres. However, these facilities may overlap in terms of the process being performed (i.e. violating *design guideline 2*). For example, a retailer may essentially perform the same process like a distributer. At the same time, different retailers may be different structurally; for example, with regards to their ordering decision. This is due to the high variance and complexity in supply chain structures.

In supply chain management, high complexity has been dealt with through buffers (Wikner & Tang, 2008). These buffers can be inventory or time. The buffers allow the supply chain to be decoupled, as long as these buffers are adequate and meet certain criteria. The different processes, before and after the buffer, can be designed separately. This gave rise to the Customer Ordering Decoupling Point (CODP) concept. CODP, also known as penetration point, is defined as

> "the point in the value chain for a product where the product is linked to a specific customer order." (Olhager, 2012: 38)

The positioning of CODP divides the supply chain into make-to-stock, make-to-order, assemble-to-order and engineer-to-order configurations. In make-to-stock configurations the stock of inventory creates a buffer between customer orders and the manufacturing process. The manufacturing process before the buffer is forecast driven rather than demand driven. An example of that is purchasing a product from a retailer (be it jeans, TV set, or car) and receiving it instantly from stock. This is in contrast to make-to-order, in which making/manufacturing the product/service is directly connected to the customer specific order. An example of that is purchasing airplane engines, custom-build computers, or high-end medical equipment. Assemble-to-order is essentially assembling the product to customer orders. The extreme opposite of make-to-stock is engineer-to-order, where the design and engineering of the product is only initiated with a customer order. For example, custom-build houses, refinery producers, or wedding-events. Figure 1 shows different possible configurations of the supply chain according to CODP position. From Figure 1, we can see that CODP divides the supply chain process to either forecast driven or demand driven processes (Olhager, 2012).

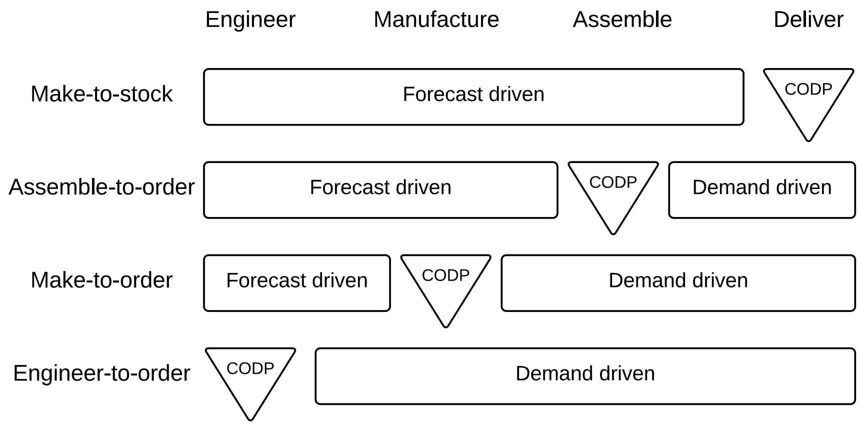

Figure 1: Different configurations of supply chains according to CODP position, based on Olhager (2003)

Wikner and Tang (2008) provide a framework dependent on CODP for closed-loop supply chains. The framework is meant to be comprehensive and encompassing all closed-loop supply chain operations. The framework divides the closed-loop supply chain into four main modules: transform to forecast (TTF), transform to demand (TTD), retransform to forecast (RTTF), and retransform to demand (RTTD). The term transform is used instead of 'make' to "avoid any implicit or explicit limitations to manufacturing since value is created in terms of both form and place." (Wikner & Tang, 2008: 350). For example, one could transform raw materials to components (i.e. manufacture); or components to assembled products (i.e. assembly). Also one could transform products from one place to another (i.e. transportation). This creates generic modules that could be parameterized depending on the process type: manufacture, assemble, transport, etc.

The framework allows the four different modules (i.e. TTF, TTD, RTTF, and RRTD) to be integrated together into different configurations to model different supply chain structures. The modules could be interchanged to represent different configurations and simulate different scenarios.

The framework satisfies *design guideline* 2; each module has a specific role and is structurally different from the other. TTF is forecast driven, and thus products are stocked at the end of the process, also the input to the module in terms of materials is forecast driven as well as the materials transformation rate. TTD is demand driven, and thus there is no stock at the end of the process, also the input to the module in terms of materials is demand driven as well as the materials transformation rate. RTTF and RTTD are essentially the same as TTF and TTD respectively, except that they resemble the reverse supply chain processes. Thus they are slightly more complex as they involve uncertainties in quality, quantity and time of returns.

These modules have similar system dynamics structures, for example: demand forecasting, orders processing, ordering design rules, and costs and revenues. This led us to take advantage of *design guideline* 4, which states that each module should have a distinct structural skeleton. The skeleton allows for the integration of mini-modules in a hierarchal manner. In that case, a mini-module is a module that is developed to be integrated within the interface of the main modules. These mini-modules are repeated structures that are used to build the main modules. For example, the TTF module can have different forecasting strategies, and thus in the TTF skeleton you could put different interchangeable forecasting mini-modules that represent the system being modelled. The same mini-module could also be used in the RTTF module. In summary, the main modules will have different structural skeletons that make use of more basic mini-modules like: forecasting, order decision rules, orders processing, and costs and revenues.

The modules interface should allow for material, information, and cash to flow between modules. However, in the context of this study, we will not consider the cash flow between modules and thus it will be disregarded. The information flows will be divided into *Outgoing Ordering Rate* and *Incoming Ordering Rate*. And the material flows will be divided into *Material Inflow* and *Material Outflow*. Thus each module essentially has the interface shown in Figure 2. Other inputs to the modules could be defined, like capacity limitation and forecasting inputs, but these inputs are not as necessary as the ones shown Figure 2. This is because the two inputs and outputs defined in Figure 2 represent the main interaction between the modules while other inputs and outputs are used as way to create flexibility in representing different systems, which satisfies *design guideline 4*. This framework provides the design rules asked for in *design guideline* 1 required for independent module development. And thus this framework provides the basis for developing system dynamics modules in supply chain.

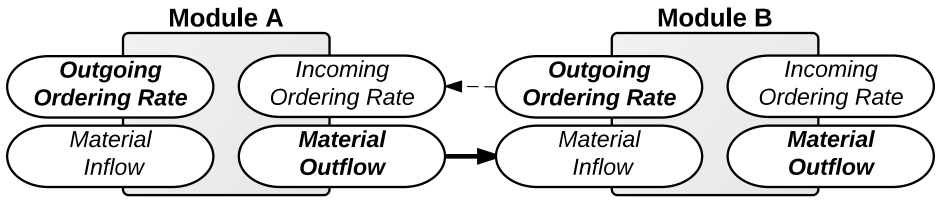

Figure 2:Main interface between two modules, where the modules' inputs are in normal text and outputs are in bold text. The dashed lines represent information flow while the solid lines represent material flow

We have developed four different system dynamics modules to be used for modelling supply chain in accordance to the theoretical framework presented above. Three of the four modules are the ones discussed above, which are TTF, TTD, and RTTD. The TTF module would be general enough to be able to represent RTTF processes as well. In addition to these I developed one extra version of TTF module, which is TTF-Push module (TFF-P). The reasoning for developing this module is the fact that sometimes material is pushed through the supply chain without ordering. For example the European Union put forward legislations, like the WEEE directive, that forces the companies to take care of their products end-of-life (Govindan, Soleimani, & Kannan, 2014). This makes companies responsible for collecting products from consumers after usage, and thus the material is pushed to the supply line once the customer decides to return the products. Also some companies decide to accept any product returns regardless of the quality and time of return in order to increase their return volume. After the material is returned the company can then control their stock of transformed material by controlling the disposal rate of excess or failed products. This is represented in the TTF-P module, which controls the stock of material by controllable disposal rather than material ordering.

In Figure 3, we present the standard version of the TTF stock and flow structure as an example of the modules developed. The figure shows the mini-modules used (grey boxes), and their inputs and outputs as well as the main stock and flow structure. The TTF module's inputs and outputs are shown as curved squares on the module frame; specifically, the module's outputs are shown in bold text.

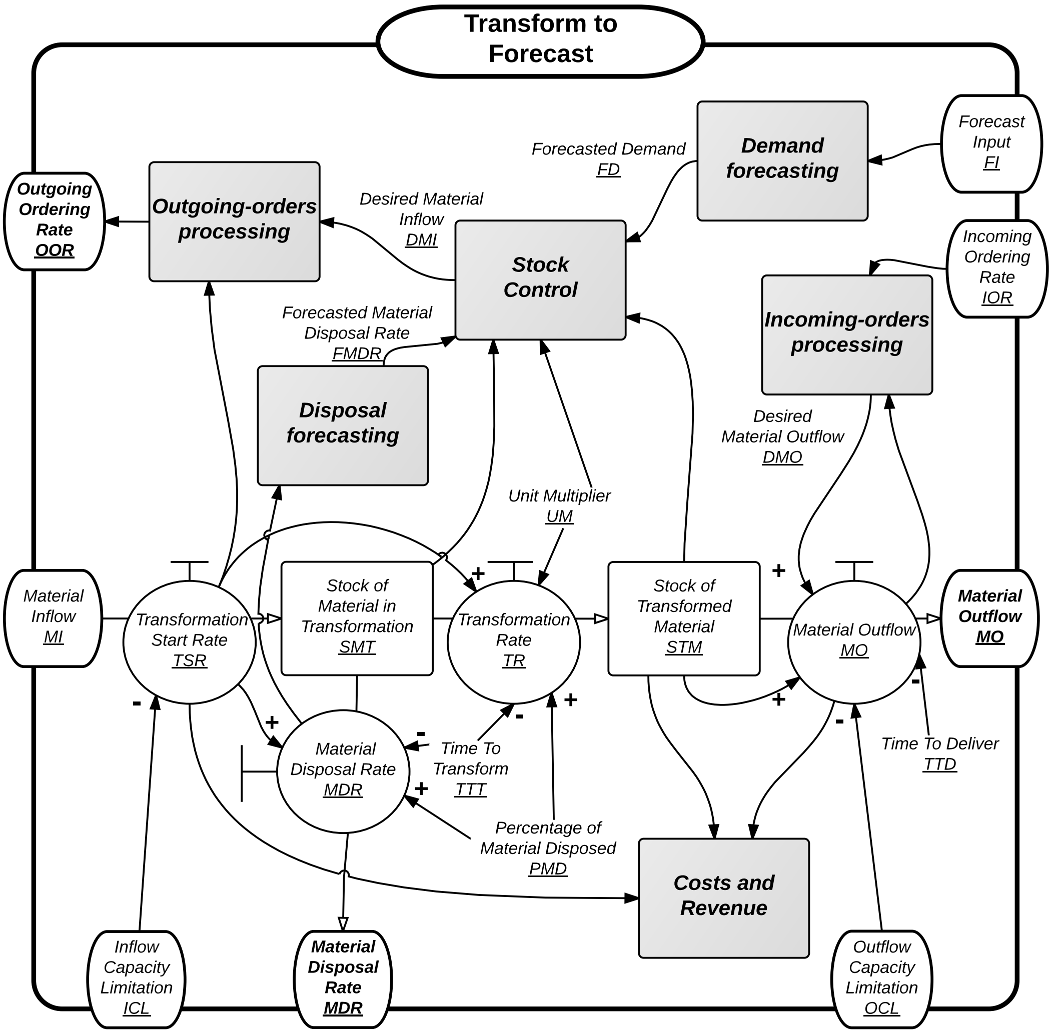

Figure 3: Transform to forecast (TTF) module general structure

In Figure 3, materials could represent things like: raw materials, components, products, livestock, or crops. The transformation process could represent things like: material extraction, production, manufacturing, assembly, maturation, or plantation. In this module the material goes through the transformation process then it is stocked in the *Stock of Transformed Material.* This stock and flow structure is very similar to Sterman's (2000: 710) Production-Inventory model. The *Incoming Ordering Rate* input is fed into the **Incoming-Orders processing** mini-module, which is responsible for determining the *Desired Material Outflow*. The **Stock Control** mini-module processes the *Forecasted Demand, Stock of Transformed material, Stock of Material in Transformation,* and *Forecasted Material Disposal Rate* to give out the *Desired Material Inflow* that is responsible for controlling the modules' stocks. The *Desired Material Inflow* is turned into *Outgoing Ordering Rate* after passing through the

Outgoing-orders processing mini-module. The *Outgoing Ordering Rate* is used as an input to another module, which then gives the needed *Material Inflow* to supply the stocks with the desired materials

In this module, the *Material Inflow* initializes the start of the transformation process; such that once the materials arrive they are directly put in the process of transformation. And so, the *Transformation Start Rate* is equal to the *Material Inflow* but is limited by the *Inflow Capacity limitation.* The *Inflow Capacity Limitation* is used to represent the capacity limitation of the transformation process. It can either be infinite, constant, or an input from a capacity module. The *Transformation Start Rate* accumulates in the *Stock of Material in Transformation,* while the *Transformation Rate* depletes the *Stock of Material in Transformation.* And so, the *Stock of Material in Transformation* represents the amount of material under transformation. After the material is transformed it is stored in *Stock of Transformed Material.* The *Material Outflow* depletes the *Stock of Transformed Material* and is determined by the *Desired Material Outflow* but limited by the available amount of *Stock of Transformed Material* and *Time To Deliver* as well as the *Outflow Capacity Limitation.* The *Outflow Capacity Limitation* acts in the same way as the *Inflow Capacity Limitation.*

There are multiple versions of mini-modules used in the main-modules; each mini-module is used to add a customizable function to the overall module. This leads to multiple possible formulations of each developed module. Although this creates flexibility in adapting the modules to different needs, it might confuse the reader. For this reason, we have developed a naming convention that will aid the reader in identifying which mini-modules are used within the specified module. Appendix B shows the usage and compatibility of each mini-module developed as well as the naming convention used in this study.

The modules developed are formulated using previously used and validated supply chain models in system dynamics as well as qualitative descriptions of the modules' functions. The TTF/TTD/RTTD modules' main stock and flow structure is well representative of a transformation process. Obviously there could be certain transformation processes that need a more detailed stock and flow structure, but this would not serve the purpose of this article. The modules developed have adequate detail and boundary to remain general yet representative to a useful degree for most transformation processes.

We have tested the module's structure and behaviour rigorously, within our capacity, to eliminate the possibility of modelling errors. In that aspect, all formulations of the modules are dimensionally consistent. Also various extreme condition tests were passed. In addition, we have performed the boundary-adequacy test, behaviour reproduction test, and behaviour prediction tests on all developed modules. Various examples of the validity tests used on the modules are presented in Appendix C.

Using the modules to represent different Supply chain configurations

Wikner and Tang (2008) identified 15 different supply chain configurations using four distinct supply processes (i.e. TTF, TTD, RTTF, and RTTD). Table 1 shows the different supply chain configurations, where any intersection between a column and a row is one possible unique supply chain configuration. Out of the 15 configurations Wikner and Tang identified nine hybrid configurations that involve an integration between at least one forecast driven process and one demand driven process (shown as roman numerals in Table 1). The forecast driven processes could consist of TTF and/or RTTF while the demand driven processes could consist of TTD and/or RTTD. For example, Configuration IX is an integration of the forecast driven processes TTF and RTTF, and the demand driven processes TTD and RTTD.

Table 1: 15 different supply chain configurations (Wikner & Tang, 2008: 358)

Demand Driven / Forecast Driven	-	TTD	RTTD	TTD+RTTD
-	-	TTD	RTTD	TTD+RTTD
TTF	TTF	I	II	III
RTTF	RTTF	IV	V	VI
TTF+RTTF	TTF+RTTF	VII	VIII	IX

Figure 4 shows how Wikner and Tang (2008) illustrate configuration IX using their framework. The customer orders decoupling point (i.e. CODP in Figure 4) separates the forecast driven supply from the demand driven supply. In configuration IX, customers can purchase a product, use it, and then either return it to be repaired/serviced (i.e.

through RTTD), return it without receiving replacement (i.e. through RTTF), or dispose it. This configuration represents the most ‘complete and integrated’ closed loop supply chain in comparison to the other configurations (Wikner & Tang, 2008). For example, in configuration V, when the customer returns a product it goes through RTTD, and if the core of the product is damaged, it is disposed to RTTF and replaced with another core from the stock of retransformed cores. If, however, the core could not be replaced, then the order cannot be satisfied. This is in contrast to configuration IX, where a newly manufactured core form TTF or TTD could replace the damaged one. Wikner and Tang (2008: 360) give the following example for configuration IX:

> “In an automotive parts remanufacturing company for example, engines cores are dismantled to obtain parts and components, which can be further used in remanufacturing. When a customer sends back an engine core and requests refurbishing, an RTTD order is released, and supplementary materials are withdrawn from the TTF and RTTF inventory at the CODP. The refurbished engine will be sent back to the original customer. In the second case (more often in this company), a customer simply identifies the model of engine in his order for a replacement, and in the meantime he will send back a core. This will release a manufacturing order (TTD). In the assembly of an engine, additional new material may be purchased (TTF) in addition to the remanufactured parts and components (RTTF) from previously returned engines.”

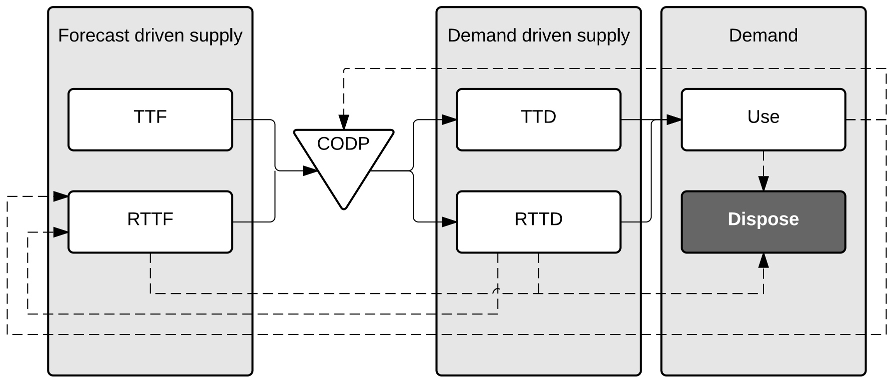

***Figure 4: Supply chain configuration* IX** (Wikner & Tang, 2008: 358)

Modelling supply configurations

In this section, we use the modules to model three different configurations for a hypothetical electronics company ‘ABC’ selling product ‘XYZ’. This is done to illustrate the modules ability to represent these configurations, and drew valuable insights from them. The data for product XYZ (i.e. product life cycle, residence time, manufacturing time, etc.) is estimated from Georgiadis et al.'s (2006) paper. The model is shown in Figure 5. In this model Company ABC manufactures components for product XYZ according to forecast (i.e. Module A). Then, product XYZ is assembled according to demand (i.e. Module B). Products are sold and returned for repair after an average *Time to Damage*. Module C receives products for repair and orders necessary components from Module A or D according to the specified ordering ratio. The ordering ratio is the ratio of components ordered from module A to components ordered from module D to supplement the repair process (i.e. material needed for module C).

In this model, we investigate the effect of changing the supply chain configuration on the total disposed products. The variable *Total disposed products* acts as a measure of resource efficiency over the lifetime of products, such that when the total disposed products are high it is an indication of loss of resources and vice versa. Changing the supply chain configuration is done by simply varying the ordering ratio. When varying the ordering ratio we determine if module C (which is RTTD) receives materials to supplement the repair process from module A only (which is TTF), module D only (which is RTTF) or both module A and D. In essence we are observing the effect of changing the *ordering ratio* on *total disposed products.*

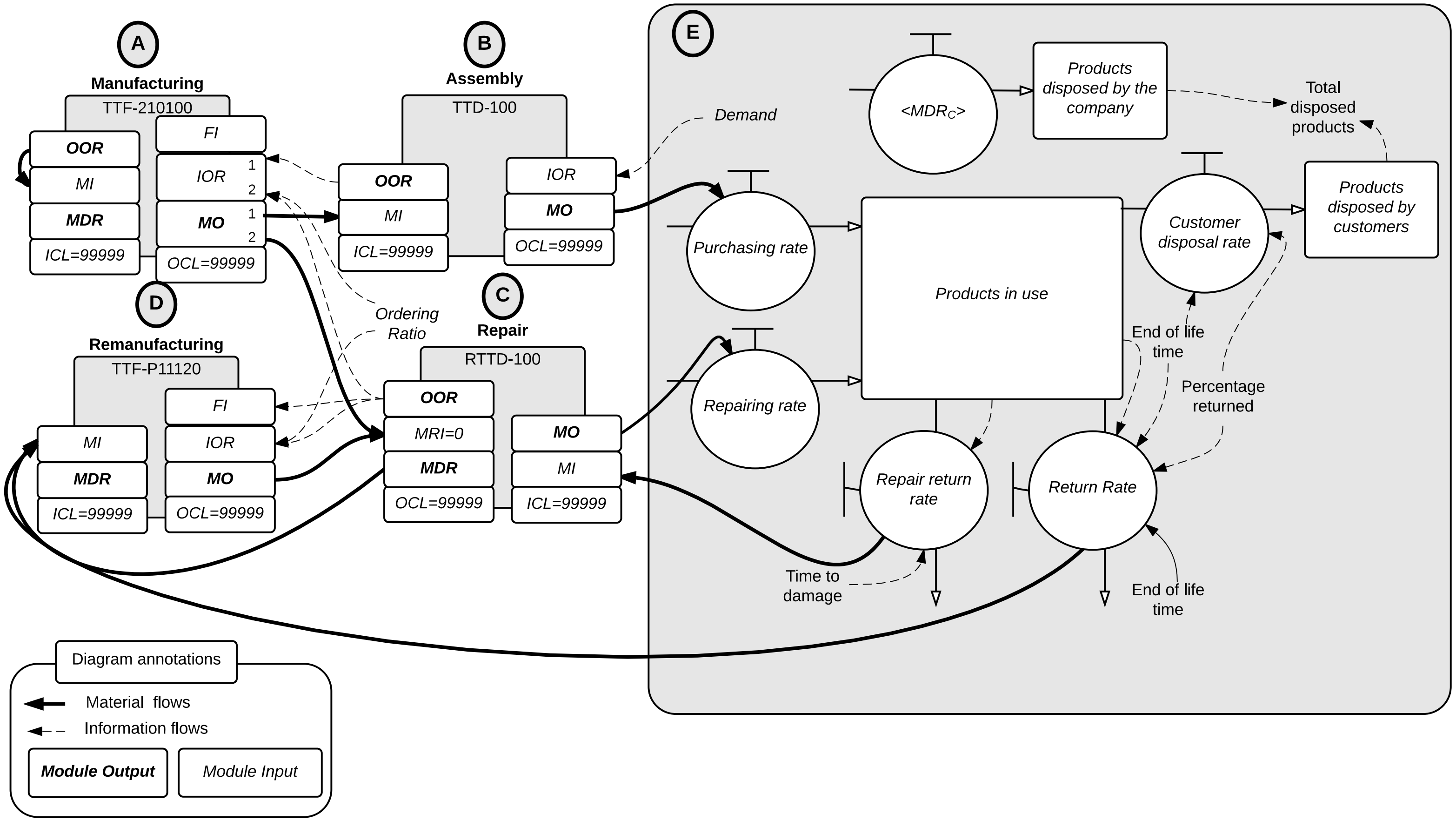

Figure 5: Closed loop supply chain model in modules

Model formulation

Module E, shown in Figure 5, is custom made for the model. The module's mathematical formulation is shown in Equation set 1. The data for the model are estimated from Georgiadis et al.'s (2006) paper and are shown in Data Table A-1 in Appendix A. It is worth noting that we assumed that components and products retain the same units, such that one component is enough to produce one product. This is done for model analysis simplification.

Equation set 1: Module E in the closed loop supply chain model mathematical formulation.

$$\mathrm{ProductsInUse}_{t+dt} = \mathrm{ProductsInUse}_{t+dt} * (\mathrm{PurchasingRate} + \mathrm{RepairingRate} - \mathrm{RepairReturnRate} - \mathrm{ReturnRate} - \mathrm{CustomerDisposalRate}) \quad (1)$$

$$\mathrm{ProductsDisposedByCustomers}_{t+dt} = \mathrm{ProductsDisposedByCustomers}_{t} + dt * (\mathrm{CustomerDisposalRate}) \quad (2)$$

$$\mathrm{ProductsDisposedByTheCompany}_{t+dt} = \mathrm{ProductsDisposedByTheCompany}_{t} + dt * (\mathrm{MDR}_{D}) \quad (3)$$

$$\mathrm{PurchasingRate} = \mathrm{MO}_{B} \quad (4)$$

$$\mathrm{RepairingRate} = \mathrm{MO}_{C} \quad (5)$$

$$\mathrm{RepairReturnRate} = \mathrm{ProdutsInUse}/\mathrm{TimeToDamage} \quad (6)$$

$$\mathrm{ReturnRate} = \frac{\mathrm{ProductsInUse}}{\mathrm{Endoflifetime}} * (\mathrm{PercentageReturned}) \quad (7)$$

$$\mathrm{CustomerDisposalRate} = \frac{\mathrm{ProductsInse}}{\mathrm{Endoflifetime}} * (1 - \mathrm{PercentageReturned}) \quad (8)$$

$$\mathrm{TotalDisposedProucts} = \mathrm{ProductsDisposedByTheCompany} + \mathrm{ProductsDisposedByTheCustomers} \quad (9)$$

Results and analysis

All the simulation runs shown below are in response to the demand shown in Figure 6. In the first run we have set the *ordering ratio* to 1, meaning that all components needed for repair are ordered from module A and non are ordered from module D. This represents configuration III in Table 1, where the supply chain has TTF, TTD, and RTTD processes. In this specific case, the *percentage returned* (i.e. the percentage of products returned by customers after products' end of life) is set to zero since there is no facility to collect customers' disposed products (i.e. no RTTF process exists).

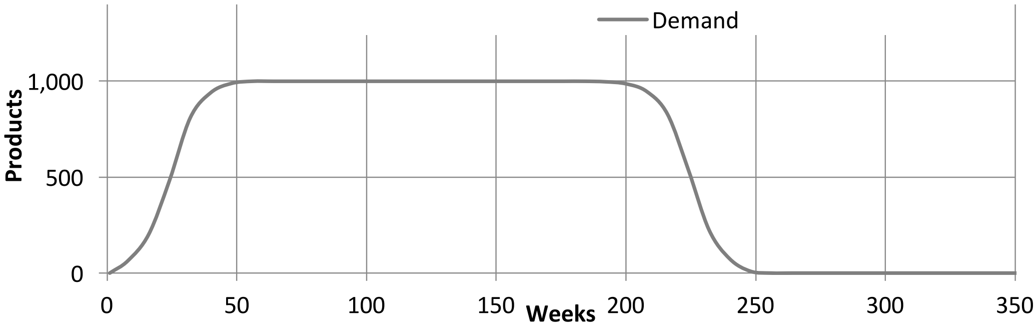

Figure 6: Time-graph used as the demand for new products in the model

Figure 7 shows the number of products disposed by the customers and the company. In the figure, there are approximately 340,000 products disposed in total, out of which the customers dispose 57% of the total; the rest is disposed by remanufacturing facilities.

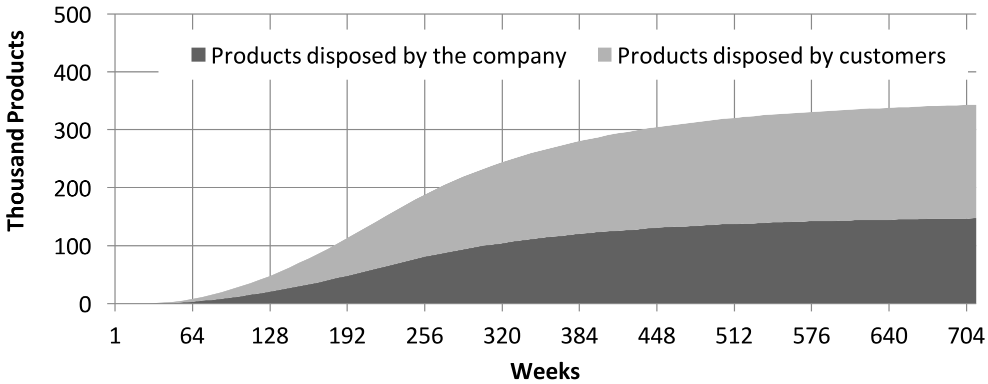

Figure 7: Number of products disposed by the company and customers for the first simulation run

It is worth noting that electronics disposed by remanufacturing facilities are more likely to be recovered faster than the ones disposed by customers. This is because companies are more likely to acquire value by selling its waste in bulk to independent third party remanufacturing and recycling facilities. In addition, governments, often, impose strict regulations for manufacturers to take care of their electronics waste in an environmentally friendly manner. The U.S. Environmental Protection Agency (EPA) reported that individual consumers have a tendency to store unused electronics longer than commercial consumers (U.S. Environmental Protection Agency, 2011). As an example, on average, individual consumers send their desktop-computers to end-of-life management facility after 12.5 years, while commercial consumers sent their desktop computers after 4.6 years (U.S. Environmental Protection Agency, 2011). Furthermore, the Consumer Electronics Association (CEA) argues the need to increase consumer awareness for appropriately disposing electronics after end-of-life; a survey conducted by CEA indicated that 58% of consumers know where to discard their electronics appropriately, and that 18% of consumers discard electronics in the trash (Consumer Electronics Association, 2014). This indicates that it is desirable to have recycling and remanufacturing facilities disposing products instead of individual consumers.

In the second simulation run, the *ordering ratio* is set to 0.5, meaning that half the components needed for repair are ordered from module A and the other half is ordered from module D. This represents configuration IX in Table 1, where the supply chain has TTF, RTTF, TTD, and RTTD processes. Figure 8 shows the number of products disposed by customers and the company when *percentage returned* equals 0 and 0.5.

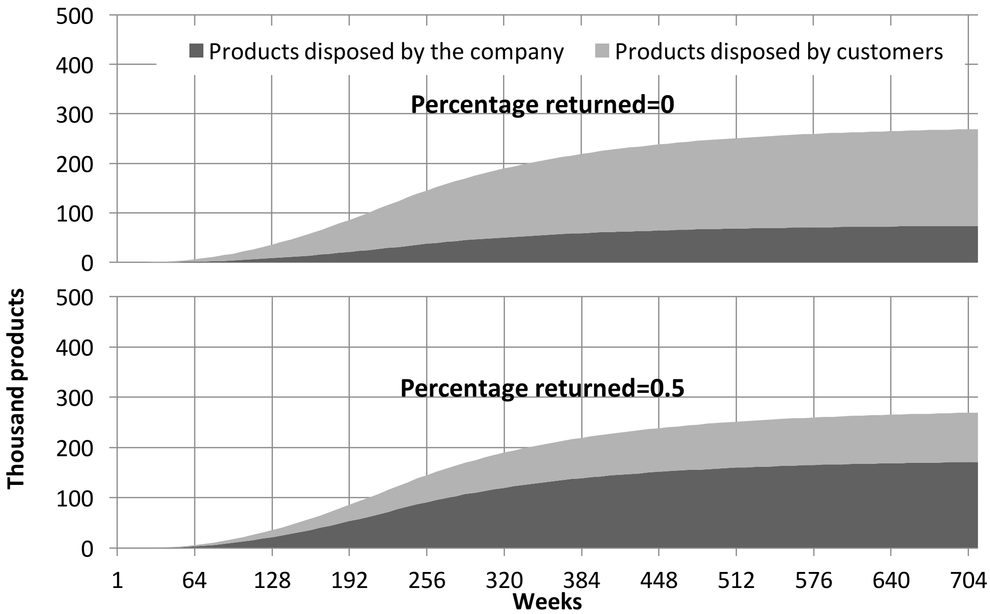

Figure 8: Number of products disposed by the company and customers for the second simulation run with percentage retuned=0 or 0.5

In the figure we see that there are approximately 270,000 products disposed in total, out of which customers dispose 72% of the total when the *percentage returned*=0 (top figure) and 36% when the *percentage returned*=0.5 (bottom figure). These two runs show that configuration IX is more efficient than configuration III in terms of material usage. Configuration IX decreased material disposal by 20% compared to configuration III. The 20% non-disposed products in configuration IX were put back into the supply chain and were remanufactured, thus increasing the lifetime value per manufactured product, and increasing the company's resource efficiency. This also goes in line with

Wikner and Tang (2008) description of configuration IX as a complete closed loop supply chain configuration. In the second simulation run, the *percentage returned* parameter had no effect on the total amount of materials disposed. However, it increased the percentage of materials disposed by the company instead of consumers; this is an expected result. As customers return more products to the company, the company disposes more products such that it is able to meet demand while minimizing storage.

In the third simulation run, the *ordering ratio* is set to 0, meaning that all the components needed for repair are ordered from module D. This represents configuration VI in Table 1, where the supply chain has RTTF, TTD, and RTTD processes. Figure 9 shows the number of products disposed by customers and the company when *percentage returned* equals 0 and 0.5.

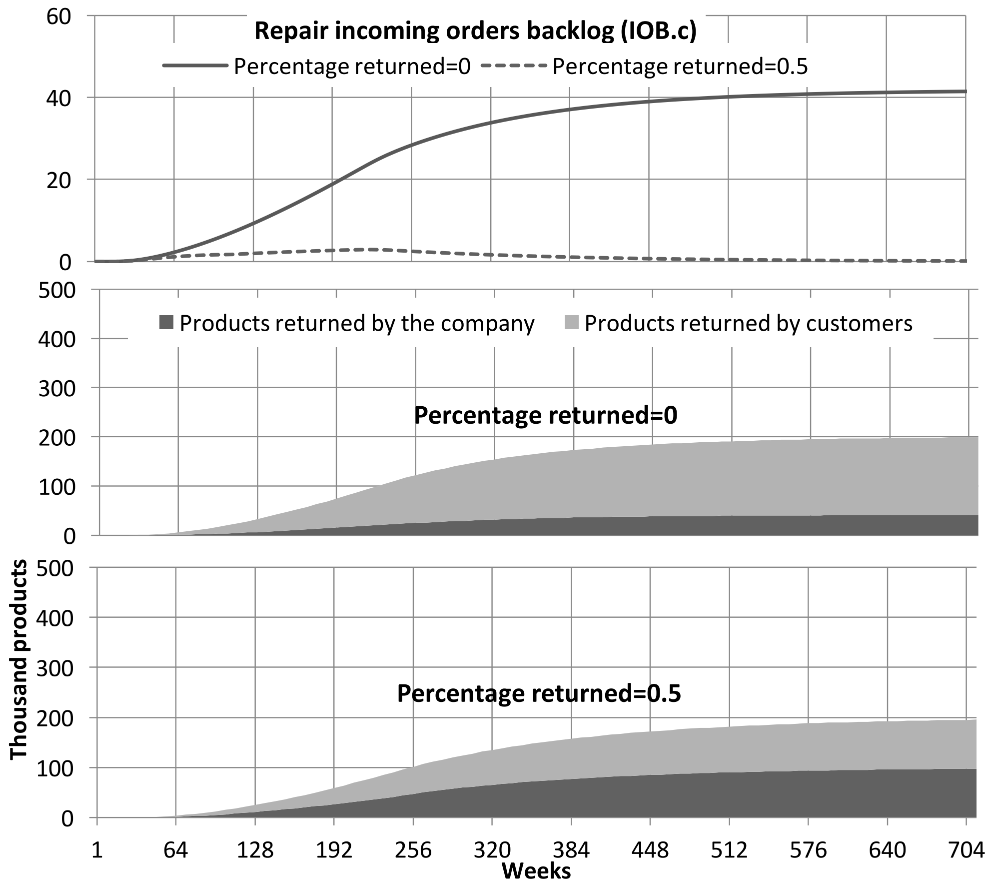

Figure 9: Number of products disposed by the company and customers and the repair incoming orders backlog for the third simulation run with percentage retuned=0 or 0.5

In the figure, we see that when the *percentage returned*=0 there are approximately 200,000 products disposed in total, out of which customers dispose 79% of the total (top figure). In addition, when the *percentage returned*=0.5 there are approximately 195,000 disposed products, out of which customers dispose 50% of the total (bottom figure). Furthermore, when *percentage returned* =0, there are 50,000 incoming orders backlog in module D (i.e. repair); this is due to the fact that there are not enough products returned to be disassembled and remanufactured to supplement repair. This confirms Wikner and Tang (2008) description of configuration VI, where RTTD orders may not be satisfied if the core could not be replaced. On the other hand, when *percentage returned=0.5* the incoming orders backlog is kept at appropriate levels due to adequate returns. This shows that such a configuration needs sufficient levels of returns. Furthermore, since all repair components are remanufactured from used products, the total products disposed decreased by 40% compared to configuration III.

The previous simulation runs show the performance of three different supply chain configurations in terms of material efficiency and orders satisfaction. The simulation results show that different configurations are suitable for different services. For example, companies with high volume of manufacturing could take advantage of configuration IX, to reduce their material consumption. Such configuration could also extend the life cycle of a product by offering repair and maintenance services. This however requires high returns, and thus companies need to increase end-of-life return awareness of their customers. On the other hand, smaller service shops that operate as a RTTD process could seek TTF/RTTF suppliers for new/used components that could be used in repair and maintenance. Furthermore, companies that have configuration VI, could offer a replacement policy through a TTD process when the core is not repairable or replaceable. For example, the customer would bring the damaged product and receive a new replacement from a TTD process, while the old product is sent back to a RTTF process. This way companies will avoid high backlogged orders shown in the third simulation run.

This model is an example of how modules could be used to represent different supply chain configurations, and draws valuable insights from simulating different scenarios. The model developed is in line with the conceptual descriptions that Wikner and Tang (2008) put forward for configurations III, VI and IX. Furthermore, in this example, we have seen how mini-modules could be easily customised to represent the system under study.

Conclusion and future research

The article was set out to develop system dynamics modules that can be used in modelling supply chain systems. Modellers are meant to use the modules by plugging them together to represent supply chain systems. The article has also sought to provide a proof of concept regarding the applications and benefits of modularity in the system dynamics field, specifically in modelling supply chains. There are similar attempts to modularity in the system dynamics field; however, there is no attempt to develop applicable system dynamics modules in the supply chain field. To pursue these goals, we followed a systematic deductive research approach where we extracted module development design guidelines from disciplines that have successfully applied the concept (e.g. software development field). The design guidelines provided us with systematic approach to modularity. We, then, used Wikner and Tang (2008) theoretical framework to apply modularity in supply chain systems. The theoretical framework categorized the supply chain processes to processes based on forecast and processes based on demand, such that the Customers Orders Decoupling Point (CODP) divide the processes based on forecast from processes based on demand. In addition, the framework differentiates forward from reverse supply chains. As such, the theoretical framework provides a complete set of processes that could be integrated together to model different supply chain configurations. Finally, we have used reoccurring system dynamics structures in supply chain modelling to develop four distinct modules. The deductive approach was used to take advantage of the vast literature on supply chains and provide a systematic approach to modularity in system dynamics that is relevant and applicable to the supply chain field as well as the system dynamics field.

This study contributes to system dynamics methodology in various ways. First, system dynamics practitioners could use the design rules specified in developing modules for different applications. The article presents a proof of concept of the potentiality of modularity in system dynamics. Second, the modules developed offer a tool for efficiently modelling supply chain systems. Experienced and inexperienced modellers could develop models to represent supply chains in less time and effort and with minimal errors. In the study, we have shown that the modules were able to replicate previously validated models. Furthermore, we have developed an intuitive understanding of the modules and their roles, which shows that when practitioners use modules frequently they can easily identify their role in the system, and intuitively understand their interactions within a complex system. Thus, the modules offer a tool to articulate complex models in simple diagrams and generate intuitive understanding of the system. Third, the modules offer a practical tool in understanding supply chain systems. For example, the modules were used to represent three different configurations of closed loop supply chains. Furthermore, when developing models from scratch, one can only change parameters to test for different scenarios, leaving the modeller with great difficulty in changing the model's configuration or overall structure. On the other hand, modules offer an easy way of reconfiguring the supply chain; plugging the modules together in different configurations offers a valuable tool in scenario analysis. This also offers a tool for adequate boundary assessment, by adding or removing modules and comparing the model's behaviour with the system's behaviour. Fourth, the innovative way of structuring the modules in a central interface with different mini-modules, encourages cumulative learning. Modellers could build and share their mini-modules, allowing others to use them and build on them. Thus, decreasing time and effort while advancing the system dynamics methodology in modelling supply chains. In addition, when modules are used in different systems, modellers could improve on them by identifying modelling or structural errors. Thus, increasing the validity of the modules, and their practical use. Furthermore, we imagine a great benefit for consulting companies using system dynamics modules in modelling supply chains. The companies could take advantage of previously developed modules and mini-modules to increase their efficiency and validity in modelling supply chains.

In summary the article has illustrated the value of modularity in the system dynamics field and, in particular, modelling supply chains. However, we came across some limitations that pave the way to future research in modularity and its applicability in system dynamics. First, the system dynamics modelling software packages are "modularly unfriendly". One of the most frustrating examples is that the software does not offer input/output plugins. As such, we recommend that system dynamics software would offer a tool in which users could develop a module, specify its inputs and outputs, and the software would automatically show the input and output nodes, so that the user could simply connect a module's output node to another module's input node.

Lastly, the modules should be arranged in a hierarchical class library, such that when a user changes a module class in the library, all the modules "under" this class change accordingly. This is extremely efficient; when a modeller identifies a structural error, he/she can change the module in the library and the software automatically changes all the modules used under this class. The second research limitation is the restrictiveness of the modules; it is not possible to represent all kinds of supply chain systems using the modules developed. This is an expected consequence since some systems could be very specific and thus difficult to represent with modules. This opens up an interesting future research direction, in which the modules developed could be used to replicate system dynamics models while testing their applicability on different supply chain systems.

Bibliography

Angerhofer, B. J., & Angelides, M. C. 2000. System dynamics modelling in supply chain management: research review. ***Simulation Conference, 2000. …***: 342–351. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=899737, January 10, 2015.

Baldwin, C. Y., & Clark, K. B. 1997. Managing in an age of modularity. ***Harvard Business Review***, (September 1997): 84–93.

Baldwin, C. Y., & Clark, K. B. 2000. ***Design Rules: The Power of Modularity Volume 1*** (illustrate.). MIT Press.

Consumer Electronics Association. 2014. Importance of Ecycling: CEA Study says More People are Recycling Electronics. http://www.ce.org/News/News-Releases/Press-Releases/2014/Importance-of-Ecycling-CEA-Study-says-More-People, May 26, 2015.

Das, D., & Dutta, P. 2013a. A system dynamics framework for integrated reverse supply chain with three way recovery and product exchange policy. ***Computers & Industrial Engineering***, 66(4): 720–733. http://linkinghub.elsevier.com/retrieve/pii/S0360835213003094, November 8, 2014.

Das, D., & Dutta, P. 2013b. Simulation study of an integrated reverse logistics in fuzzy environment. ***Lecture Notes in Electrical Engineering***. http://www.scopus.com/inward/record.url?eid=2-s2.0-84872521717&partnerID=40&md5=03b5e023d754164984af8f7c619907d4.

Forrester, J. W. 1961. ***Industrial Dynamics***. http://books.google.nl/books?id=4CgzAAAAMAAJ, Wiley.

Forrester, J. W. 1968. ***Market Growth as Influenced by Capital Investment***. http://books.google.nl/books?id=BtKunQEACAAJ, Industrial Management Review.

Forrester, J. W. J., & Senge, P. P. M. 1980. Tests for building confidence in system dynamics models. ***TIMS Studies in the Management Sciences***, 14: 209–228. http://www.sdmodelbook.com/uploadedfile/32_0f93d902-b52f-4b1f-a658-5a5d144c3d1b_Tests for Building Confidence JWF Senge.pdf, January 13, 2015.

Foster, I. 1995. ***Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering***.

Georgiadis, P., & Athanasiou, E. 2010. The impact of two-product joint lifecycles on capacity planning of remanufacturing networks. ***European Journal of Operational Research***, 202(2): 420–433. http://linkinghub.elsevier.com/retrieve/pii/S037722170900352X, January 9, 2015.

Georgiadis, P., & Besiou, M. 2008. Sustainability in electrical and electronic equipment closed-loop supply chains: A System Dynamics approach. ***Journal of Cleaner Production***, 16(15): 1665–1678. http://linkinghub.elsevier.com/retrieve/pii/S0959652608001091, September 9, 2014.

Georgiadis, P., Vlachos, D., & Tagaras, G. 2006. The Impact of Product Lifecycle on Capacity Planning of Closed-Loop Supply Chains with Remanufacturing, 15(4): 514–527.

Gibb, A. G. F. 2001. Standardization and pre-assembly- distinguishing myth from reality using case study research. ***Construction Management and Economics***, 19(February 2015): 307–315.

Govindan, K., Soleimani, H., & Kannan, D. 2014. Reverse logistics and closed-loop supply chain: A comprehensive review to explore the future. ***European Journal of Operational Research***, 240(3): 603–626. http://linkinghub.elsevier.com/retrieve/pii/S0377221714005633, July 17, 2014.

Hines, J. 1996. Molecules of Structure. https://www.vensim.com/documentation/Models/Molecules/molecule.pdf.

Kamath, N. B., & Roy, R. 2007. Capacity augmentation of a supply chain for a short lifecycle product: A system dynamics framework. ***European Journal of Operational Research***, 179(2): 334–351. http://linkinghub.elsevier.com/retrieve/pii/S0377221706002141, December 8, 2014.

Mall, R. 2009. ***FUNDAMENTALS OF SOFTWARE ENGINEERING***. http://books.google.no/books?id=CFhtgKrKL1EC, PHI Learning.

Olhager, J. 2003. Strategic positioning of the order penetration point. ***International Journal of Production Economics***, 85: 319–329.

Olhager, J. 2012. The role of decoupling points in value chain management. In H. Jodlbauer, J. Olhager, & R. J. Schonberger (Eds.), ***Modeling value***: 37–47. Physica-Verlag Heidelberg.

Poles, R. 2013. System Dynamics modelling of a production and inventory system for remanufacturing to evaluate system improvement strategies. ***International Journal of Production Economics***, 144(1): 189–199. http://linkinghub.elsevier.com/retrieve/pii/S0925527313000704, December 24, 2014.

Richardson, G. P. 2011. Reflections on the foundations of system dynamics. ***System Dynamics Review***, 27(3): 219–243.

Sterman, J. 2000. ***Business dynamics : systems thinking and modeling for a complex world***. Boston: Irwin/McGraw-Hill.

Sterman, J. D. 1989. Modeling Managerial Behavior: Misperceptions of Feedback in a Dynamic Decision Making Experiment. ***Management Science***, 35(3): 321–339.

U.S. Environmental Protection Agency. 2011. ***Electronics Waste Management in the United States Through 2009***. http://www.epa.gov/epawaste/conserve/materials/ecycling/docs/fullbaselinereport2011.pdf.

Vlachos, D., Georgiadis, P., & Iakovou, E. 2007. A system dynamics model for dynamic capacity planning of remanufacturing in closed-loop supply chains. ***Computers & Operations Research***, 34(2): 367–394. http://linkinghub.elsevier.com/retrieve/pii/S0305054805001000, January 8, 2015.

Wikner, J., & Tang, O. 2008. A structural framework for closed-loop supply chains. ***The International Journal of Logistics Management***, 19(3): 344–366. http://www.emeraldinsight.com/doi/abs/10.1108/09574090810919198, January 9, 2015.

Wolstenholme, E. F., & Coyle, R. G. 1983. The development of system dynamics as a methodology for system description and qualitative analysis. ***Journal of the Operational Research Society***, 34(7): 569–581.