Simulation Modeling Unit: A Bridge Cross–Domain Communication in Building and Developing Simulation

หน่วยสร้างแบบสถานการณ์จำลอง: สะพานสำหรับการเชื่อมผ่านข้ามโดเมน ในการสร้างและพัฒนาแบบจำลองสถานการณ์

Dr. Kitti Setavoraphan
Lecturer in MBA Program
Faculty of Business Administration
Panyapiwat Institute of Management
E-mail: kittiset@pim.ac.th

Abstract

For a real-world problem, numerous simulation models have not only been developed by individuals but also built on different simulation modeling languages or environments. By a means of reproducing those simulation models, it is very difficult for afterward modelers to apply those simulation concepts (e.g., entities, attributes, processes, etc.) for their specific purposes related to the same problem. As a consequence, the conceptual simulation modeling (CSM) plays a key role in translating the problem's concepts into a standard simulation modeling framework – which can be reused to develop simulation models on different languages or environments. This paper is focused on applying one of the CSM tools, named "Simulation Modeling Unit (SMU)" to bridge a gap of cross-domain communication between problem (application) domain and simulation domain. An illustration of development of a CSM for lockage operations using the SMU will also be given.

Keywords: Conceptual Simulation Modeling, Inland Waterway Transportation Systems, Lockage Operations

บทคัดย่อ

สำหรับปัญหาๆ หนึ่งที่สามารถหาคำตอบด้วยวิธีการสร้างแบบสถานการณ์จำลองนั้น พบว่ามีแบบสถานการณ์จำลองจำนวนมากที่ถูกพัฒนาขึ้นโดยวิธีแตกต่างกันไม่ว่าจะเนื่องด้วยนักพัฒนาหรือภาษาหรือสิ่งแวดล้อมสำหรับการสร้างแบบสถานการณ์จำลอง แต่ถ้าจะมุ่งเน้นในเรื่องของการนำแบบสถานการณ์จำลองเหล่านั้นมาใช้ใหม่เพื่อหาคำตอบให้กับปัญหาที่ใกล้เคียงกับปัญหาเดิมซึ่งมีลักษณะเฉพาะมากขึ้น ดูเหมือนจะเป็นสิ่งที่ยากสำหรับนักพัฒนาคนต่อๆ มากับการนำแนวทางการสร้างแบบสถานการณ์จำลองเดิมมาใช้ เพราะฉะนั้นวิธีการสร้างแบบสถานการณ์จำลองทางความคิดที่ใช้สนับสนุนการสร้างแบบสถานการณ์จำลองจึงเข้ามามีบทบาทมากยิ่งขึ้น อันเป็นแนวทางที่สำคัญในการสนับสนุนการนำกระบวนการ แนวคิด และรูปแบบของสถานการณ์จำลองเดิมกลับมาใช้ใหม่ได้อย่างมีประสิทธิภาพมากยิ่งขึ้น ในบทความนี้ จะมุ่งเน้นในการนำเสนอเครื่องมือที่ถูกพัฒนาเพื่อการสร้างแบบจำลองทางความคิดที่ใช้สนับสนุนการสร้างแบบสถานการณ์จำลองเพื่อลดช่องว่างการถ่ายโอนความคิดระหว่างปัญหากับแบบสถานการณ์จำลอง โดยยกตัวอย่างกรณีการจำลองสถานการณ์การปฏิบัติการประตูกั้นน้ำที่ใช้ยกหรือลดระดับน้ำในแม่น้ำเพื่ออำนวยความสะดวกในการเดินทางของเรือบรรทุกสินค้า

คำสำคัญ: การสร้างแบบจำลองทางความคิดที่ใช้สนับสนุนการสร้างแบบสถานการณ์จำลอง ระบบการขนส่งทางน้ำ ปฏิบัติการประตูกั้นน้ำที่ใช้ยกหรือลดระดับน้ำในแม่น้ำ

Introduction

Among a number of simulation models developed and applied in Modeling and Simulation (M&S), it is seen that one problem domain can be solved within a similar simulation modeling framework–using different structural and behavioral characteristics. Prior to a variety of modeling and translating methods, those simulation models have been created on particular simulation modeling languages or environments. It seems to be fine unless there is a need for reproducing or reusing the simulation concepts in a new study for a similar problem domain.

There is an issue that has broadly been discussed among new-entry simulation modelers (e.g., university students, research assistants) and even experienced ones. Their only concern is how to initiate a simulation project when assigned to deal with the same problem domain for specific requirements. For example, the assignment is to develop a simulation model for a lockage operations system to do feasibility studies for constructing and operating locks on the Chao Phraya River, Thailand. They probably either develop their own simulation models (which is time-consuming) or reuse the existing models (which is convenient but difficult to do so). Either ways, they still find difficulties in translating the problem domain into proper simulation concepts.

Likewise, to develop a totally brand-new simulation model is not reasonable, especially by inexperienced simulation modelers or non-domain experts (of lockage operations systems). The main reasons are: (1) a risk that the model can be run but not valid; (2) another risk that the project will never be completed. Obviously, these kinds of risks always happen in such large, complicated simulation projects.

Back to the reusable approach, only one concern for those modelers to be cautious is that none of the simulation models can be perfectly reusable. Though, it is possible to interpret some previous simulation studies, for instance, the Fox River locks system with SLAM (Bandy, 1987; 1988; 1991), the Illinois waterway system with ProModel (Bandy, 1996), and the Ohio River and the Panama Canal waterways with AutoMod 11.0 (Biles et al., 2004) into, e.g., simulation concepts, parameters, and logical processes. It is indeed difficult to apply those resources into the new assignment, due to the boundaries of cross-domain communication (Arons, 1999). This well reflects to the impacts of the barriers between simulation languages or environments by a means of simulation contents and contexts.

The key solution for this dilemma is to create an interface that can facilitate the transformation of concepts from problem (application) domain into simulation domain during simulation modeling processes. The interface has been well-known as a conceptual simulation model (CSM) (Benjamin et al., 1997). A CSM can be developed through a pattern-based and knowledge-based approach (Zhou et al. 2005). This approach allows the modelers to capture (encapsulate) structural and behavioral characteristics of the target system into a form of logical and descriptive representations. Based on the approach, a CSM tool called "Simulation Modeling Unit (SMU)" has been developed to facilitate construction of one or more simulation modeling instances that contain process description, node reference number, attribute, and operation aspects for building conceptual simulation models (Setavoraphan, 2005).

In this paper, a demonstration of building a CSM for a simple lockage operations system by using the SMU is presented. The remaining parts of the paper are organized as follows: In Section 2, the processes of building a CSM are briefly described. Transformation of the CSM to a Visual SLAM model is shown in Section 3 with model descriptions. Conclusions and discussions are given in Section 4.

Processes of Conceptual Simulation Model Building

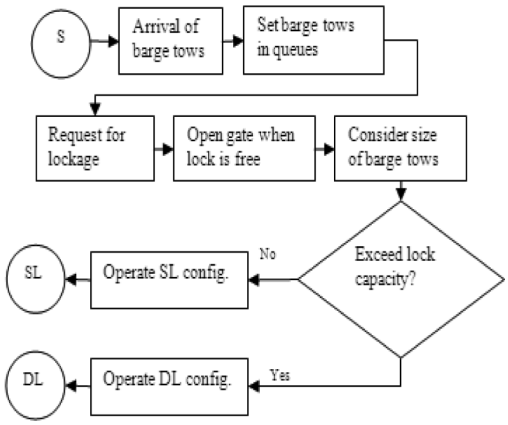
In an inland waterway transportation system, one or more locks are required when a stretch of river is made navigable by bypassing an obstruction such as a rapid, dam, or location on the river where there is a significant change in elevation. Not only are the locks used to make a river more easily navigable but also to allow a canal to cross country that is not level. Canals associating with locks system provide connections between rivers and other waterway either upstream or downstream of the elevation change (Bandy, 1996). As a fixed chamber whose water level can be varied, a lock can raise of lower vessels between stretches of water at different levels. For further details of general lockage operations, see the U.S. Corps Army of Engineers' official website and Bandy's works (1987, 1988, 1991, and 1996).

In this section, the concepts of the inland waterway lock systems, including lockage operations (derived from above) are transformed and represented in terms of conceptualization representations. There are five processes for conceptual simulation model building.

1. *Capture system descriptions*: First of all, the modelers must be able to obtain all the required and necessary knowledge relating to the target system and to understand its concepts. The methods of knowledge acquisition can be reviewing literatures, interviewing domain experts, and doing research on the real lockage operations systems.

2. *Set objectives for simulation modeling*: Wise selection of the processes and objects of interests in the model is very helpful for the modelers to establish model boundaries–leading to a well-designed scope for simulation modeling goals. For example, this study is set to illustrate how a lock operates and controls the traffic on the Chao Phraya River when barge tows and their tow boats arrive and request for travelling through it.

3. *Translate descriptions into a logical process-flow model*: The main purpose is to represent a sequence of activities and decisions taking place in the system. The more levels the system can be decomposed into, the more details its activities can be described. Figure 1 shows a logical process-flow model of the system in general (0^{th} level of decomposition).



Note: S= Start, SL= Single Lockage, DL= Double Lockage

Figure 1: An overview of a sub logical process-flow model for the lock system

4. *Identify simulation modeling elements*: To reduce difficulties in translating logical models into implementation models, the simulation modeling concepts need to be identified as the following categories (Zhou et al., 2004):

a) *Entities:* Objects flow through the system to receive services provided by a sequence of activities. A barge tow consisting of a set of barges and a tow boat is defined as an entity for this study.

b) *Resources:* Objects are placed at fixed locations of a system according to certain configurations to provide a means of services. Each lock has two resources which are its chamber and electric wrenches.

c) *Functional activities:* An activity is a logical process transformed into a simulation function. A collection of simulation functional activities contains processing entities, controlling entities flows, and collecting generated data from the flows in the system, which can be illustrated as follows:

- Create entities: Barge tows are created and flow through the system;

- Assign entities: An entity's attributes are, for example, arrival time, number of barges, and travelling directions;

- Process entities: Lockage is defined as a delay-time process – that deals directly with the entities' travelling in the system;

- Store/hold entities: Arrival barge tows requesting for lockage are held in a waiting queue;

- Aggregate/disaggregate entities: A barge tow is a set of barges and a tow boat, which is batched into a single file and unbatched when needed;

- Transport/route entities: Entities are moved along the routes through the system;

- Branch the flow of entities: Conditions are made for distributing entities according to their directions and number of barges;

- Control the movement of entities: Barge tows send and receive signals for locking through. Signals are generated to control the traffic at the lock;

- Dispose entities: Barge tows leave the system when lockage is completed; and

- Collect data/statistics: Time in system and waiting time in queues of each entity are collected for analysis. Utilization of electric wrenches is also determined.

d) Input and output requirements: Time between arrival rates, service rates, and delay times are inputs. The outputs are expected to be, for instance, number of barges (for either SL or DL) and time in the lock system.

5. Build a conceptual simulation model: From most simulation modelers' points of view, the processes described in 1-4 seem to be appropriately sufficient for them to build a simulation model on a specific purpose. However, there still exist some amounts of the modelers that find difficulties in developing a simulation model. Integrating simulation modeling concepts to create simulation at implementation level requires experiences in modeling and acknowledgment in particular simulation languages or environments. To reduce the gaps between cross-domain knowledge requirements and the modelers' skills, a conceptual simulation model (CSM) is exploited as an interface that bridging them together.

Based on the ideas of Simulation Modeling Unit (SMU) approach, a CSM is developed through the transformation of knowledge representations in a platform of process descriptions associating with object-oriented and pattern-based approaches. Under these approaches, each logical activity (process) can be viewed as an object class that contains attributes and operations relating to simulation modeling elements – which later functions as a simulation modeling instance (aka. a simulation modeling unit). Figure 2 presents general structure of an SMU.

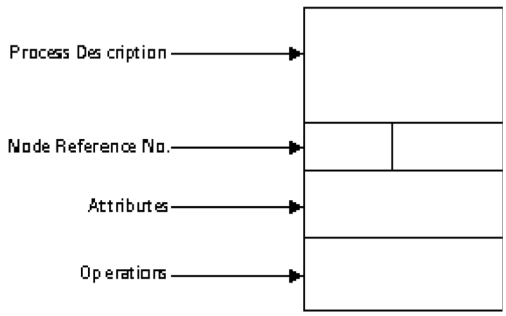


Figure 2: A simulation modeling unit

Each SMU consists of process description, node reference number, attribute, and operation aspects. First, process description displays a logical activity existing in the problem domain. Second, node reference number is used as the reference numbering scheme to identify the uniqueness of each SMU and to define levels of decomposition. Third, each attribute provides information of a structural model element performing an action and containing one or more sets of data values required for both entities and resources to apply in simulation. The final aspect, operations, is referred as an initial function activating each SMU to accomplish its descriptive processing behavior. Each operation includes one or more sets of functional elements similar to those in simulation.

Furthermore, the SMU is able to deploy other object-oriented features such as polymorphism (e.g., methods and procedures) and aggregation (e.g., a-part-of relationship for decomposition and specialization). These features help enhance capabilities in constructing simulation by a means of software engineering development, which leads to more dynamics and agilities in transforming domain concepts into simulation concepts. See Setavoraphan's work (2005) for more details about the basics and applications of SMU.

As shown in Figure 3, a CSM is built by a set of SMUs to represent the general lockage operations in simulation concepts (that are transformed from Figure 1).

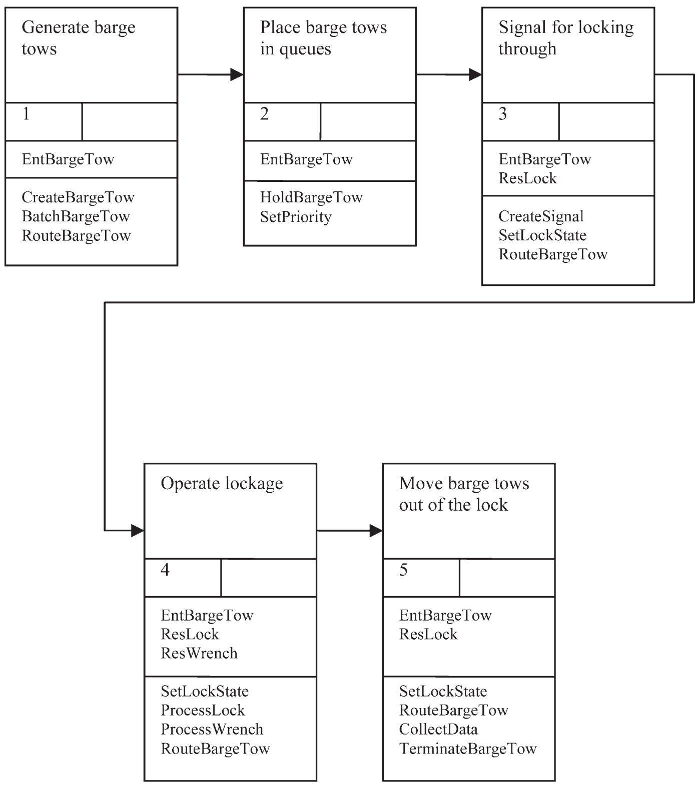
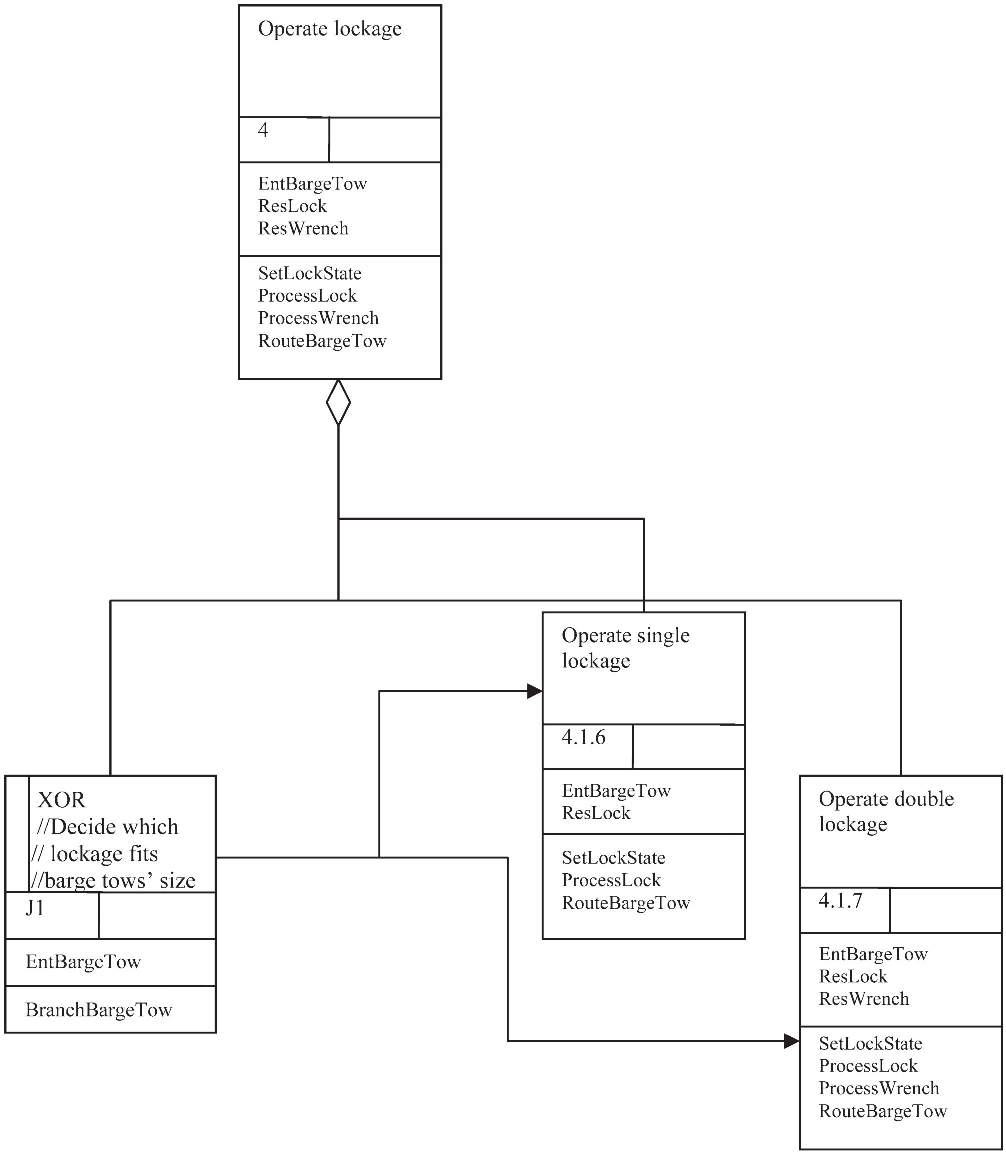


Figure 3: A CSM for lockage operations

Besides, the node reference# 4 can be decomposed into two sub lower-level of the lockage operations: single and double lockage, corresponding to barge tows' size (dimensions) for locking through. Figure 4 shows the first level decomposition of the SMU *Operate Lockage*.



Note: SMU *XOR J1* is used as a junction corresponding to conditional branching.

Figure 4: 1st decomposition of the SMU *Operate Lockage*

Sample descriptions of the attributes and operations are given in Table 1 and 2, respectively.

Table 1: Descriptions of attributes

Attribute Name	Description
EntBargeTow	A barge tow is defined as an entity flowing in the lock system. A barge-tow entity consists of a set of barges and a tow boat. It contains entity type, direction, and number of barges.
ResLock	A lock is a resource that takes an action in filling or draining water, depending on whether a barge-tow entity travels up or down. The lock resource has its own name, resource#, activity time, and capacity.
ResWrench	An electric wrench is a resource used to pull out the barges completing the first lockage. The wrench resource includes resource# and activity time.

Table 2: Descriptions of operations

Operation Name	Description
BatchBargeTow	An action is to accumulate a set of barge tow that requests for locking through.
BranchBargeTow	Each barge-tow entity is routed according to conditions.
CreateBargeTow	A barge-tow entity is created with a set of barges and a tow boat.
CollectData	Statistics data of interests are collected for analysis
HoldBargeTow	Barge-tow entities are held in queues waiting for signals to enter the lock
ProcessLock	The lock is operated for lockage by a means of delay-activity times.
ProcessWrench	Electric wrenches are used when DL is required.
RouteBargeTow	Each barge-tow entity is routed for specific destinations
SetLockState	An action (of sending a signal) verifies a status of the lock (busy/idle).
TerminateBargeTow	Each barge-tow entity is terminated when it leaves the system.

Building a CSM is determined as repetitive processes, demanding for a lot of revisions and feedbacks–to make it logical, feasible, and valid. However, it is important for the modelers to be able to realize where their boundaries (aka. satisfactions or understandings) end. Also, remember that a CSM is just used to facilitate the development of simulation models–not a magic wand that can create them immediately. The next step is to transform the CSM to a simulation language or environment to build a simulation model. Visual SLAM, a simulation language, has been selected as well as the destination of the transformation. (see Pritsker and O' Reilly, 1999 for understanding Visual SLAM).

Model Descriptions

Development of Visual SLAM network representations of the lock system involves with the use of logic, entities, resources, variables, and various network nodes, corresponding to the simulation concepts and elements described in the previous section. Flows of entities through the network are controlled by activities defined by either durations or conditions. Add-ons modification is needed for making the network model more agile and effective. Following examples are given to demonstrate how to transform those SMUs into the Visual SLAM network nodes.

As shown in Figure 5, the SMU *Generate Barge Tows* (from Fig. 3) can be transformed into a set of Visual SLAM network nodes, whose required framework is to create a function that generates a number of barge-tow entities traveling either upstream or downstream on the river – with random time-between arrivals defined in XX[1] and XX[2], respectively.

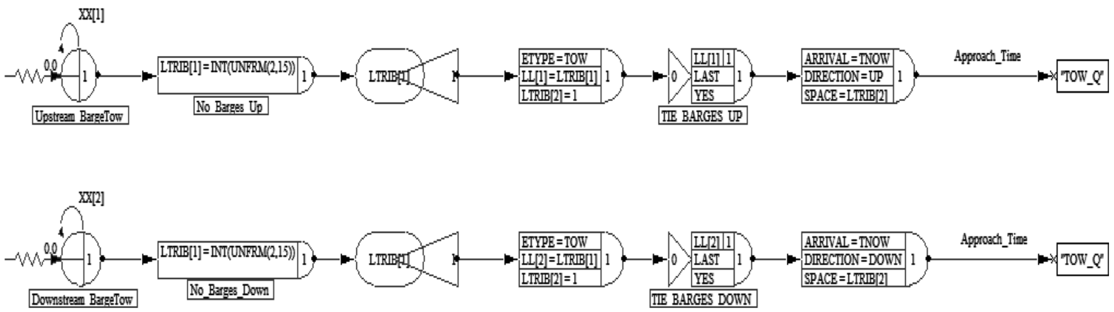


Figure 5: Creation of barge tows going either upstream or downstream

Actually, a barge tow carries one or up to fourteen barges plus a tow boat. An UNBATCH node is used to create random numbers of barges plus a tow boat defined in LTRIB[1] for each entity created from a CREATE node of each direction. The number of identical entities released from the UNBATCH node are assigned attributes to specify entity type (1 = row), identity group number (LL[1]), and size (LTRIB[2]), which are grouped together into a single file by a BATCH node. Three attributes are assigned to each batch to define its arrival time (ATRIB[1]) at the current simulation time (TNOW), traveling direction (1 = upstream, 2 = downstream), and required space for the lock (LTRIB[2]). The batch is routed to a waiting queue for barge tows that request for locking through with specified approaching-time duration.

Not every SMU can be directly transformed into the target simulation language or environment. Likewise, for Visual SLAM, it sometimes requires the modelers to configure network nodes to reflect to the requirements described within the frameworks of one or more SMUs. This indeed demands the modelers' experiences in each particular simulation language or environment. Figure 6 is a good example for this case.

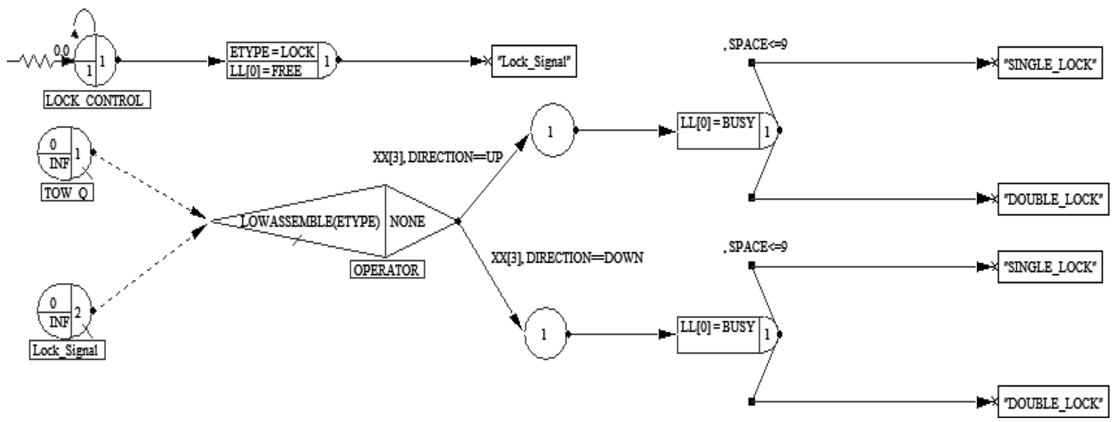


Figure 6: Creation of lock-traffic control signal and operation of lock-control

Figure 6 shows the network that represents traffic control at the lock, corresponding to a means of the SMU *Place Barge Tows in Queues* and the SMU *Signal for Locking Through*. A signal-control entity for locking through is created from a CREATE node, which is assigned its entity type (e.g., lock) and sent to a waiting queue for releasing "enter" signal. The status of lock (LL[0] = 0) is defined "free" before the lock signal is sent to the waiting queue. A SELECT node with ASSEMBLE option is used to control traffic at the lock by waiting each of the two entities for being available at the queues and combining them together to create a barge-tow entity permitted for locking through. The assembled entity maintains its attributes of the barge-tow entity (e.g., direction and space) to be used to select its routes during flowing through the network. It travels with a duration time specified in XX[3] to enter the lock. The lock status is then changed to "busy" (LL[0] = 1). Next, determination of lockage operations is made due to the barge tow's required space. If the space is less than or equal to 9, the entity is routed to single lockage operation. Otherwise, double lockage is operated, instead.

Moreover, it is seen that Figure 6 also includes some partial concepts of the SMU *Operate Lockage* in terms of selection of lockage-operation type. This shows how important the modelers' flexibility in translating concepts from knowledge representations like SMUs into a simulation language or environment such as Visual SLAM is needed. It is found out that the modelers must be able to define the boundaries of representing structural and behavioral characteristics within a well-designed framework of simulation contexts and contents–leading to modularization of Visual SLAM network nodes. Not only can each module reflect to the concepts specified in individual SMUs but also affect to its and others' functionality and continuity when composition takes place. Therefore, management of levels of details (e.g., decomposition) of conceptualization is crucial. The more the details of concepts are given (to define where the boundaries are located), the better the designs of modules are made (to identify in which those Visual SLAM network nodes are grouped).

As of applying methodologies of decomposition and modularization, the modelers are able to layout individual modules' scopes, flows, and relationships, which helps reduce or eliminate complexity in modeling. For instance, it is fine for the modelers to transform the SMU *Operate Lockage* into only one set of Visual SLAM network nodes that include both lockage operation types. However, the result might turn out as a complex module that would create errors – depending on the modelers' experiences and expertise. Figure 7 and 8, thus, are the examples of taking advantages from both decomposition and modularization.

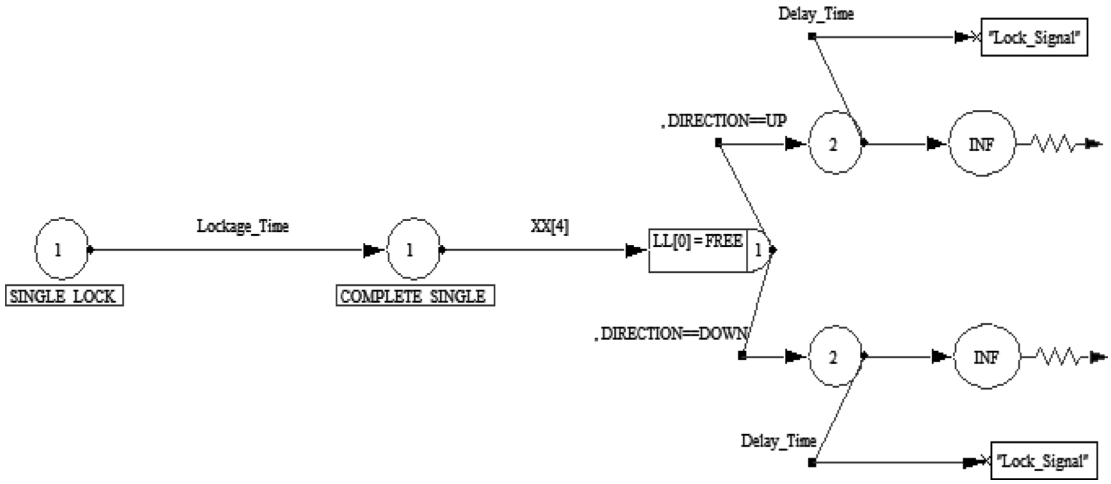


Figure 7: Single lockage operation

The module given in Figure 7 shows the general processes for a single lockage operation. In this study, a lockage operation is determined as a time-consuming activity, so the main focus of designing this module is to find out how long each barge tow spends in lockage time. Lockage time is defined as a duration time that a lock takes for filling or draining water into or out of the lock. When the lockage is completed, the barge tow moves out of the lock with XX[4] duration time. Then, the status of lock is turned to be "free" again. The barge tow continues its trip following its designated direction and leaves the system a TERMINATE node. The lock signal is sent back to its waiting queue with a particular delay time.

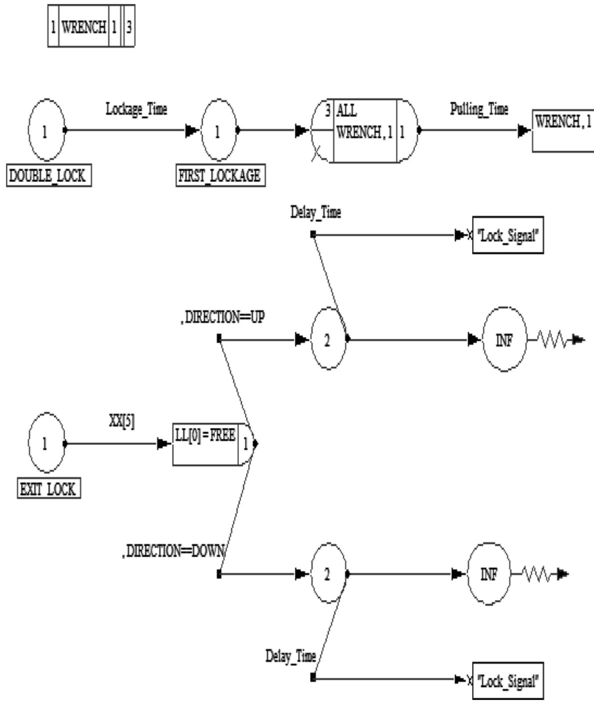


Figure 8: Double lockage operation

As well, Figure 8 illustrates another module representing a double lockage operation, which is focused on not only lockage time but also resource-utilization time. In general, after the first lockage is completed, a wrench is utilized for pulling the barges in and out of the lock for locking through – that consumes time for each pulling activity. A RESOURCE block is used to identify resource WRENCH that has a capacity of 1 to be used in the module. An AWAIT node is used for the barge-tow entity representing the front part as it waits for being pulled by the WRENCH resource. When the WRENCH resource is available, it takes some pulling-time duration to move the part out of the lock. A FREE node is used to free one unit of resource WRENCH immediately, so that it is available for the next use. The water elevation inside the lock then returns to the initial level to allow the entity, at this time, representing the back part to enter the lock for the second lockage. Both operations take place within specified lockage times. XX[5] represents a delay time for the back part to move out of the lock and combine together with the front part. LL[0] is changed to be 0 to turn the lock status to be free. The barge tow leaves the system to continue its journey at a TERMINATE node. The lock signal is delayed for a period of time before backing to its queue.

All the modules of Visual SLAM network nodes illustrated above can be further improved for better performance as well as the SMUs. As stated earlier in this paper, building a CSM is iterative processes, which is also applied to the processes of creating a simulation model.

At the modeling satisfaction level, both SMUs and simulation modules will be standardized as references for future studies in the same problem domain. Therefore, degrees of reusability depend on how well they perform to answer those questions in modeling.

Conclusions

This paper represents an application of Simulation Modeling Unit (SMU) to develop a conceptual simulation model (CSM) with a purpose of facilitating construction of simulation models such as the lockage operations system with Visual SLAM as demonstrated (Note: Another demonstration, the regional distribution center systems with Arena, can be found in Setavoraphan's thesis (2005)). It is seen that the simulation modeling concepts and elements can be captured and formalized into conceptualization level. This helps the modelers implement those into their simulation models.

Moreover, the SMUs can be reused for building simulation models (within the same problem domain) on any other simulation languages or environments. It creates better performance in communications between modelers and between concepts and languages via effective and efficient representations. However, the SMUs are still too abstract for the modelers to build simulation when the system is more complex and has various constraints. Thus, having a team with a variety of individual skills plays the key role in reducing or eliminating difficulties in managing complexity.

Another issue to be included is that like many other modeling languages, SMU has been considered as an in-house modeling language–that is only used within a closed community or organization. This implies that a mutual agreement among team members in deploying SMU to communicate their simulation-work flows is indeed required–due to their different modeling backgrounds. To reduce gaps among theirs, a number of trainings and practices are essential.

References

Arons, H. D. S. (1999). Knowledge-based modeling of discrete-event simulation systems, *Proceedings of the 1999 Winter Simulation Conference*.

Bandy, D. B. (1987). Simulation of Fox River locks boat traffic, *Presented at St. Louis ORSA/TIMS Joint National Meeting*.

Bandy, D. B. (1988). Fox River locks SLAM simulation model, *Proceedings of the 1988 Winter Simulation Conference*.

Bandy, D. B. (1991). Simulation of Fox River locks boat lift, *Proceedings of the 1991 Winter Simulation Conference*.

Bandy, D. B. (1996). Simulation of the Illinois waterway locks system, *Proceedings of the 1996 Winter Simulation Conference*.

Benjamin, P. C., Blinn, M., Fillion., and Mayer, R. J. (1993). Intelligent support for simulation modeling: A description-driven approach, *Proceedings of the 1993 Summer Computer Simulation Conference*.

Biles, W. E., Sasso, D., and Bilbrey, J. K. (2004). Integration of simulation and geographic information systems: Modeling traffic flow in inland waterways, *Proceedings of the 2004 Winter Simulation Conference*.

U.S. Army Corps of Engineers. (2006). Navigation. Retrieved October 5, 2006, from: *http:// www.swl.usace.army.mil/navigation/ lock.html*.

Pritsker, A.A.B. & O'Reilly, J.J. (1999). Simulation with Visual SLAM and AweSim. New York: John Wiley.

Setavoraphan, K. (2005). Conceptual simulation modeling for regional distribution center systems, *Master's thesis, Indiana State University, Terre Haute, Indiana, USA*.

Zhou, M., Setavoraphan, K., and Chen, Z. (2005). Conceptual simulation modeling of warehousing operations, *Proceedings of the 2005 Winter Simulation Conference*.

Zhou, M., Son, Y. J., and Chen, Z. (2004). Knowledge representation for conceptual simulation modeling, *Proceedings of the 2004 Winter Simulation Conference*.

Dr.Kitti Setavoraphan received his Ph.D. in Industrial Engineering from the University of Oklahoma, USA in 2009. He also earned his Master of Science in Industrial Technology from Indiana State University, USA in 2005 and Bachelor of Computer Engineering from Assumption University, Thailand in 2002. He is currently a full-time lecturer in the Master Business Administration program at the Faculty of Business Administration, Panyapiwat Institute of Management. His research interests are in the areas of systems modeling and analysis, simulation, supply chain and logistics management, operations and production management, and risk management.