

SIMULATION DATA EXCHANGE (SDX) IMPLEMENTATION AND USE

Dave Sly

Director of e-Factory Business Development
UGS e-Factory
2321 North Loop Drive
Ames, IA 50010, U.S.A.

Shreekanth Moorthy

e-Factory Product Manager
UGS e-Factory
24800 Northwestern Highway
Southfield, MI 48075, U.S.A.

ABSTRACT

Facility layouts of existing or proposed production systems often form the basis of a dynamic simulation study to prove the design concept and to determine if the required throughput (jobs per hour) can be achieved. Traditionally, this requires the simulation engineer to duplicate the CAD drawing in the simulation environment before being able to model it. With sophisticated manufacturing systems such as conveyor networks, material handling systems, automotive plants and powertrain facilities, this job can be very difficult, time consuming and error prone. An application and interface was needed to provide an automated integration of the layout and simulation technologies for manufacturing. This paper discusses SDX (Simulation Data Exchange) and its implementation.

1 INTRODUCTION

AutoCAD from Autodesk is the predominant application used globally for creating factory layouts. As a result, FactoryCAD[®] was created by UGS to enhance the intelligence of these AutoCAD-based layouts.

UGS has developed a number of intelligent CAD objects for facilities design. These objects allow the facility designer to design and modify the facility layout with minimal effort and tedious drafting duplication. These objects include everything from conveyors, racks and cranes, to tables, containers, fencing and mezzanines.

These objects are embedded with simulation relevant data such as Time to Fail, Time to Repair, Cycle times, Scrap rates, Conveyor Speeds, Conveyor junctions and merges, AGV and Aisle paths, et cetera. An intelligent extraction routine exports object details such as type of object (machine, conveyor, buffer, etc.), object location, and other physical parameters, along with all simulation relevant data that had been previously embedded into the objects from the layout into a standardized XML text file. This file eliminates the need to recreate physical and even some run control information in the dynamic simulation package. This file has

become what is a common data format called Simulation Data Exchange—SDX (Moorthy 1999) that can serve as an input to the automatic generation of discrete event simulation models. The simulation model then generates animation data that is subsequently available for dynamic viewing and analysis in several simulation and dynamic viewing tools.

Currently several DES vendors, such as AutoMod[®], WITNESS[®], Simul8[®], etc can read, and in some cases even create, SDX files that can serve as an input to the automatic generation of discrete event (DES) simulation models.

A dynamic link with a FactoryFLOW[®] database allows the SDX file to have part information such as part numbers, routing, individual part cycle times, scrap rates, and set up times. Thus, the SDX file now has a complete set of organized data from both, the CAD model, as well as, a database that contains material routing, and other manufacturing related data.

The FactoryCAD drawings can be either a cartoon or a detailed full-scale representation of equipment. Apart from being intelligent, these objects may be viewed as a 2-D drawing or a 3-D model (both may be displayed simultaneously) to facilitate further refinement or modification.

2 SIMULATION DATA EXCHANGE (SDX) FILE DESCRIPTION

The SDX file is used as input information (along with other generic input data externally maintained), to generate and run a simulation model of either the entire layout represented on the drawing or a specific windowed area. The file contains header information specifying drawing source data, simulation model units, run control and shift information.

Following the header section of the SDX file, any number of objects is defined. Some of the types of objects are PART, CONVEYOR, OVERHEAD POWER & FREE, VEHICLE, MACHINE, BUFFER, PATH NETWORK, P&F_NETWORK. Apart from objects that have an actual geometric shape, there are other objects like SHIFT, RUNTIME, and STATISTICS. Depending on the type of object, a number of related “details” describe all of the at-

tributes of the object necessary for the simulation model. The objects are listed one after the other, as the extraction routine traverses through the layout. Within each object, that is, between every [OBJECT BEGIN] and [OBJECT END] object parameters and other stored values (TTF, TTR, Setups, Cycle Times, Cost Data, etc.) are listed against descriptive field names. This makes it very easy for the translation tool to read the SDX file and build a model.

Table 1: Object Types Within the Layout Drawing Containing Simulation Relevant Information

-
1. LOAD
 - Active
 - Passive
 2. SHIFT
 3. OVERHEAD POWER AND FREE CONVEYOR
 - Overhead Accumulating
 - Inverted Accumulating
 4. MONORAIL
 - Overhead Non-Accumulating
 5. FLOOR CONVEYOR
 - Non-Accumulating
 - Accumulating
 6. CROSS TRANSFER CONVEYOR
 - Non-Accumulating
 - Accumulating
 7. VERTICAL LIFT
 - Non-Accumulating Floor Conveyor Type
 - Non-Accumulating EMS Type
 8. PIVOT
 - Non-Accumulating
 9. ELECTIFIED MONORAIL SYSTEM
 - Overhead Accumulating
 10. TRANSPORTER
 - Fork Truck
 - Automated Guided Vehicle
 - EMS Transporter
 - Tugger
 11. PATH
 12. AISLE
 - General
 - Automated Guided Vehicle
 - Pedestrian
 - Departmental
 13. CARRIER
 - Inverted Power and Free Carrier
 - Overhead Power and Free Carrier
 - Monorail Carrier
 14. BUFFER
 - Last In First Out
 - First In First Out
 - Restore Sequence
 15. MACHINE
 - Batch Operation Machine

- Assembly Machine
 - Production Machine
 - Single Operation Machine
 - Multi Station
16. RUNTIME
 - No Animation
 - Partial Animation
 - Full Animation
 - Normal
 - Trace
 17. PART ROUTING DATA
 18. ASSEMBLY TREE
 19. OBJECT TO OBJECT CONNECTION
 20. WORKCENTER
 21. STATION ON COVEYORS AND MACHINES
 - Load Station
 - Unload Station
 - Process Run Station (load does not stop)
 - Process Stop Station (load stops)
 - Idle station
 - Decision station
 - Marriage
 - Disassembly
 - Assembly
 22. OPERATOR
 - Self Relief
 - Tag Relief
 - Mass Relief
 - Material Handling
 - Maintenance
 - Setup
 23. DOCK
 24. STATION_AREA
 - Main Line Station
 - Off-Line Sub-assembly
 25. INDUSTRIAL

Example: The following flow chart shows the flow of, let's say PART_A from the Dock to Ship. After Machine 2, the 10% gets reworked at machine Rework. Of the reworked parts, 70% get scrapped, and 30% flow back into the line at Machine 3.

Table 2: - Example Part Routings Showing Branching of Parts

PART_A, DOCK, MACH_1, 1, 1, 1, 1, 0, NULL, NULL
PART_A, MACH_1, MACH_2, 1, 1, 1, 1, 0, NULL, NULL
PART_A, MACH_2, REWORK, 0.1, 1, 1, 1, 0, NULL, NULL
PART_A, MACH_2, MACH_3, 0.9, 1, 1, 1, 0, NULL, NULL
PART_A, REWORK, MACH_3, 0.3, 1, 1, 1, 0, NULL, NULL
PART_A, REWORK, SCRAP, 0.7, 1, 1, 1, 0, NULL, NULL
PART_A, MACH_3, MACH_4, 1, 1, 1, 1, 0, NULL, NULL
PART_A, MACH_4, SHIP, 1, 1, 1, 1, 0, NULL, NULL

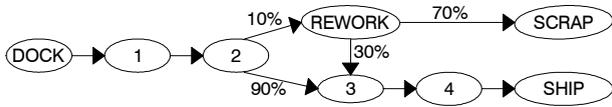


Figure 1: Example Process Flow Diagram Showing Branching of Parts

3 SDX EXTRACTION IN XML FORMAT

Previously SDX information was written out in a plain ASCII text file format. With the recent release of FactoryCAD 7.0, SDX information is saved in an XML format. This new format conforms to all the specifications that are defined in the SDX standard of versions later than version 4.2. In addition, the existing user interface for SDX extraction is provided with a modification.

The only visible difference between the plain ASCII text file and the XML file is the way the tags are displayed. The ASCII file might have a tag called [OBJECT BEGIN] and [OBJECT END], while the XML type SDX file will have <OBJECTother data..... </OBJECT>

With XML being an already acceptable, extensible standard of writing information, the transformation of plain ASCII tagged SDX files into XML tagged SDX files was only natural.

4 READING THE SDX FILE INFORMATION INTO FACTORYCAD®

A new capability has been added to FactoryCAD® 7.0 program to read an SDX file. The data file will needed to be in the XML format as defined by SDX versions later than 4.2. Only non-graphical object data will be read into the layout. That is, data such as positions of equipment, equipment sizes, et cetera will not be read in. It is also assumed that equipment Name, Type, Internal Type, and CAD ID will not be different from the ones in the drawing.

An object found in the SDX file but not found in the drawing will be ignored. The software will check for existence of equipment as read in from the SDX file by verifying the integrity of the names and their respective CAD IDs. The user interface requests the user to choose an SDX file to be and then will bring up a dialog that lists all the objects that are common to the drawing and the SDX file. The associated SDX information being read has the capability to update any external references (XREF) that are attached to the main drawing. This function will operate correctly if, and only if, the SDX file being read was initially written out with the “Add Drawing Name as Prefix” option. This way, when the SDX file is being read the program knows what data is associated with which object. A checkbox is provided beside each of the objects in the list. This checkbox allows the user to choose whether or not that particular object’s information needs to be updated. Two buttons are provided; “Select All” and “Clear All”.

5 THE SDX PROCESS

As shown in Figure 2 below, SDX information can be manually input into the SDX file, imported in, or derived from a default file. After extraction, the model is to be translated into a new simulation model, or reflected as an append or modification on an existing model. Appropriate changes in the data can be made, and recommendations in the form of a newly extracted SDX file can then be re-imported back into the FactoryCAD layout model.

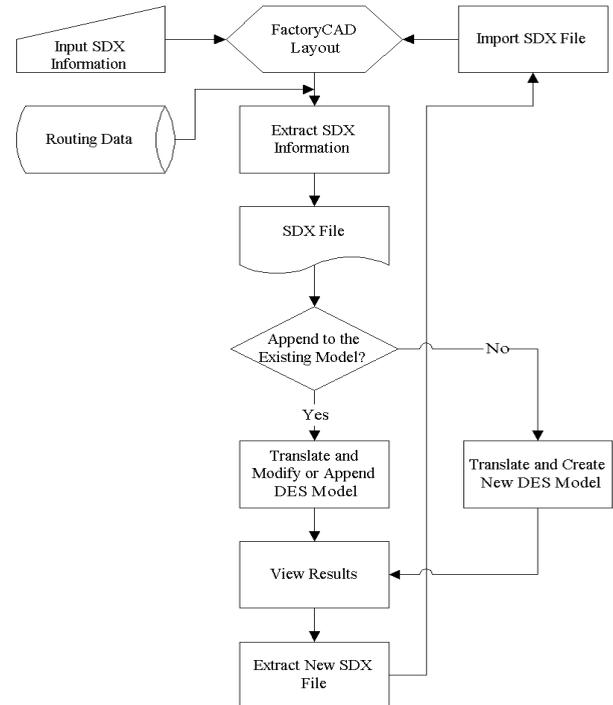


Figure 2: A Base Level SDX Process

6 EDITOR FOR SDX INFORMATION

Inspired by SDX implementation business processes, an external editor was created to allow editing of SDX information external to the layout. Typically, the layout engineer would extract an SDX file, which is then externally edited via the SDX Editor, and then, the SDX file is read back into the layout. With this process, layout work and SDX work can proceed in unison. Figure 3 depicts this process.

The navigator-oriented dialog box interface is a new addition for SDX users in order to simplify the editing of SDX information, as shown in Figure 4. This easy-to-use XML editor application provides users with the ability to select an XML object from the navigator on the left and then view and edit all appropriate attributes to that object on the right. Object types are classified, color coded and symbolized in the navigator list box to simplify the identification and editing process.

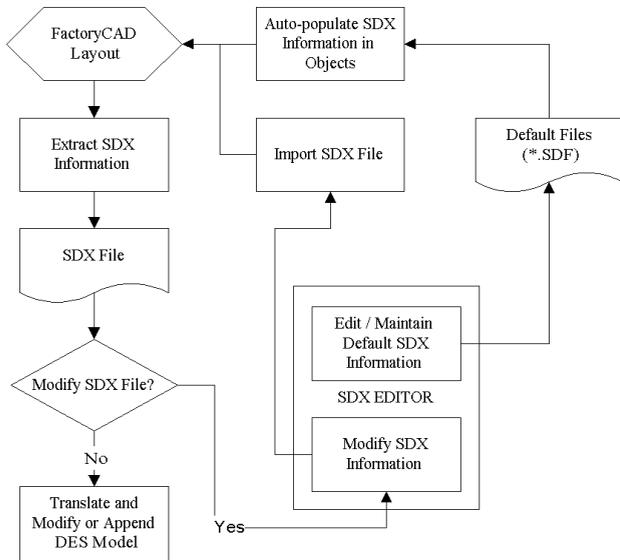


Figure 3: SDX Editor Process

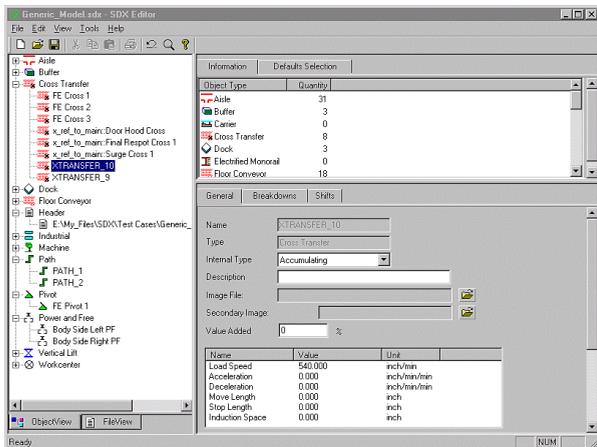


Figure 4: Main Screen of SDX Editor

Furthermore, the user is not be able to add or delete objects from the SDX file. This editor provides only a means to access the SDX information external to the drawing, and modify non-graphical data.

The SDX editor behaves in the following manner:

- The user chooses an SDX object from a pull down menu or tabs. eg.: Machine_1
- The user then proceeds to modify requisite data.
- Finally, the user clicks on a button to update.
- Iteratively, the user can choose objects and modify the SDX information.

7 MAINTENANCE OF DEFAULT SDX DATA

The basic purpose of the default editing capability is to allow users to maintain and update their default information where-

from data can be derived into either new objects being created in the layout, or the objects being modified in the existing SDX file. A separate file is maintained in an XML format, not unlike a standard SDX file, that contains all of the non-graphical default information of machines, conveyors, and other objects that carry SDX information. Graphical information such as positions of equipment, equipment sizes, et cetera are not included. Each FactoryCAD® layout file contains a field that references the path and file name of the Default file.

The default editor is very much similar to the SDX editor; indeed, it's a part of the same application. However, the default editor allows for objects to be added and deleted. Also, the editor behaves in the following manner:

- The user chooses a FactoryCAD® object from a pull down menu. eg.: Machine / Conveyor
- The user then proceeds to enter a name and other requisite data. It is to be noted that the name the user enters here is not extracted in the SDX file. This is simply a means of identification for the objects to derive information from.
- Finally, the user clicks on a button to update.
- Iteratively, the user can choose objects and create multiple defaults.

The SDX / Default editor is a stand-alone program. In other words, the user will not be required to have FactoryCAD® loaded on his or her computer to use this application.

The following is a list of new FactoryCAD® features that have been implemented to support the Default Editor:

- If a default file is associated with the drawing, the defaults will be used.
- When the user creates a new object, he or she can choose a name from a pull down menu, wherefrom data is populated in the SDX fields of the object. If default information does not exist for a particular object type (eg.: Overhead Power and Free conveyor), no information will be populated.
- The user can then modify the data fields if need be.

8 FUTURE DEVELOPMENT: ANIMATION VISUALIZATION – AUTO-GENERATE PATHS IN FACTORYVIEW™

Once the simulation model has been generated, in all or part, as a result of the layout plan, it is only logical to define a standard to feed the trace file results of the dynamic object movements generated by the DES model back to an animation of the layout be defined. This step will require a synchronization of the layout elements, both dynamic and static, forward to the trace file itself. It is also imperative that the coordinate system of the layout be scaled appropriately, and accurately enough, to provide adequate positioning in the layout environment.

In order for this capability to be universally applicable, a standard, possibly an extension of SDX, needs to be developed to provide an independent definition of dynamic elements of the layout over time. This standard also needs to take into account color changes, graphical representation swapping, and textual/attribute displays. In addition, it may also be necessary for the graphical environment, and even possibly the animation system, to address local analysis issues such as graphical queuing on conveyors, conveyor intersection management and the dynamic details of human movement (such as walking, picking, placing, etc).

9 SUMMARY

SDX is an evolving and practical standard that is finally bringing the activities of factory and material handling design to those of simulation and analysis. As an independent standard, recently managed and coordinated by NIST, SDX has the necessary support and foundations to revolutionize the way in which factories, manufacturing facilities and material handling systems are engineered for far into the future.

REFERENCES

- Moorthy, S. 1999. Integrating the CAD Model With Dynamic Simulation: Simulation Data Exchange. In *Proceedings of the 1999 Winter Simulation Conference*, ed. P. A. Farrington, H. B. Nembhard, D.T. Sturrock, and G.W. Evans, 276-280. Phoenix, Arizona.
- Moorthy, S. & D. Sly. 2000. New Technology Merges Layout Planning and Simulation, Proceedings of North American Material Handling Show 2000, Detroit, MI.

AUTHOR BIOGRAPHIES

DAVE SLY is Director of e-Factory Sales at UGS. He has a Bachelors and Masters degree in Industrial Engineering and an MBA from Iowa State University. Dave is a registered professional engineer in the State of Iowa and is a Senior member of the Society of Manufacturing Engineers, Institute of Industrial Engineers and the Society for Computer Simulation.

SHREEKANTH MOORTHY is a Product Manager at the e-Factory group at UGS. He has undergraduate degrees in Mechanical and Industrial Engineering. He attended Louisiana Tech University, where he obtained a M.S. in Industrial Engineering. He has eight years of experience working in plants involved in manufacturing heavy engineering products and in the automotive industry. His research interests lie in the areas of production optimization, simulation, scheduling, material flow, and facilities planning. His e-mail and web addresses are <Shreekanth.Moorthy@ugs.com> and <www.ugs.com>.