

REDUCING MODEL CREATION CYCLE TIME BY AUTOMATED CONVERSION OF A CAD AMHS LAYOUT DESIGN

Igor Paprotny

Automation Planning and Design
Group
PRI Automation, Inc.
Mesa, AZ 85208, U.S.A.

Wendy Zhao

Software Division
PRI Automation, Inc.
Billerica, MA 01821, U.S.A.

Gerald T. Mackulak

Department of Industrial
Engineering
Arizona State University
Tempe, AZ 85287, U.S.A.

ABSTRACT

Simulation is a popular tool for accurately estimating the performance of an automated material handling system (AMHS). Accuracy of the model is normally dependent on a detailed description of the AMHS physical system components and their coordinate positions. In this paper, a methodology is defined for automatically inputting the physical system components used to describe an AMHS within a simulation language. The method is based on data extraction from a CAD layout file of the system. Automatically generating the physical system components reduces simulation model building time and increases model accuracy.

1 INTRODUCTION

The design of any AMHS requires precise estimation of its capabilities. In semiconductor industry, material handling typically accounts for 10%-15% of the total manufacturing costs. Needless to say, it is important to design a system that operates as close as possible to the optimum. A single AMHS component or subsystem can cost from \$20,000 to \$500,000. Inaccurate performance estimation can quickly lead to a non-competitive AMHS design.

Simulation is often used as a tool for evaluating the performance of the proposed AMHS, because of its ability to accurately model the dynamics involved with AMHS operation. However, the simulation approach requires the creation of a valid and dimensionally accurate model that reflects the behavior of the proposed system. The creation of these models is often time consuming. Even though several simulation packages today provide model templates that diminish some of the coding effort, inputting the individual layout characteristics necessary for an accurate model, is still a time-consuming ordeal.

A CAD layout of the proposed AMHS is always created prior to simulation analysis. This is because AMHS

designs are usually based on an imposed set of physical constraints, because it is often beneficial to perform static validation and optimization of the design prior to simulation, and finally, installation requires a blueprint diagram. It is thus convenient to overlay a provided (CAD electronic) facilities layout with the AMHS design.

Static validation and optimization of the design prior to simulation can reduce the number of iterations necessary for it to reach optimality. EAI's FactoryFlow™ software is an excellent static evaluation tool for capacity calculations of the AMHS. The program is built on top of AutoCAD, and thus requires input of information in the form of a CAD system layout.

Finally, the output of the design process has to include a layout representation of the proposed system. CAD is preferred as the standard layout creation and storage method, thus the CAD AMHS layout representation has to be created before any equipment can be installed. In fact, engineers do not feel comfortable until they see a "layout" of the proposed system in the building footprint. Figure 1 illustrates a typical example of an AMHS layout. Note the amount of detail present in even an average system layout.

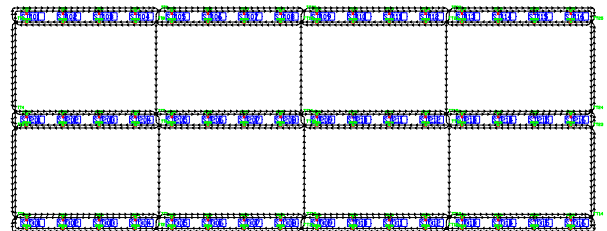


Figure 1: A Typical CAD AMHS Layout

2 PROBLEM STATEMENT

In order to reduce the model building cycle time, it is suggested that an automatic conversion scheme could be implemented to translate an AMHS CAD layout into a

simulation model. The available approaches to facilitate the model conversion were found insufficient from a cycle time reduction standpoint. For example, AutoMod™ v. 8.6 (Phillips 1998) has an option to import a CAD layout as a static background template to be used for model creation. The model creator is however still required to manually trace the model components on top of the imported layout, thus the method does not provide any significant reduction in the model building effort.

This paper presents the generic methodology developed for the purpose of automating the creation of AMHS simulation models. A specific example of employing the conversion methodology is presented, and issues regarding its implementation and operation are discussed.

3 CAD TRANSLATION METHODOLOGY

Any CAD layout can be categorized as a collection of entities and attributes containing information about the system. The set of information is usually complete regarding the spatial location of the system components (x,y,z), but incomplete with respect to operational characteristics such as speed, flow rate etc. The conversion process involves extracting relevant data from the layout, enhancing it with additional information, converting it to a format that can be used by the simulation software and exporting it to the simulation model.

It is important to remember however, that the algorithm will extract only predefined and expected components from the CAD layout. The layout must be created using a set of standards that can later be recognized by the extraction software. Because of the amount of effort involved in the development of the algorithms and CAD standardization, the automation of the model creation will in general pay off only if a larger number of models is to be created using the same method and a common set of components. This condition is highly applicable to AMHS designs, where subsequent models are created based on a defined set of components.

In general, the conversion process can be divided into two steps, A and B, as illustrated in figure 2. Step A requires an interface method with the CAD layout that allows for extraction of layout-entity attributes. This can be done either internally within a CAD package, or externally by accessing the layout file.

Currently available CAD packages provide several solutions applicable to both extraction methods. AutoCAD allows for internal implementation of LISP and Visual Basic subroutines that can be used to search and export the characteristics of relevant drawing entities. Dassault Systems Microstation™ has the ability to perform an

intelligent filter-export function that can export relevant entities into a customized text-file.

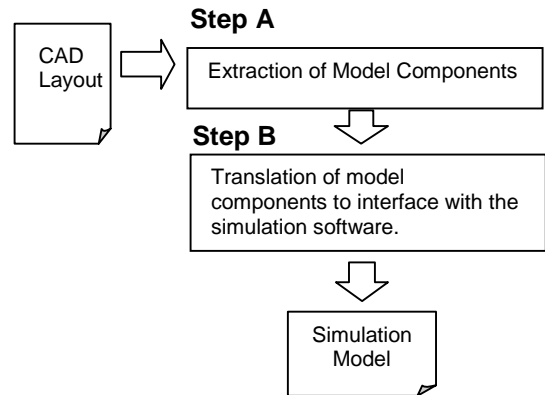


Figure 2: A Generic CAD to Simulation Model Conversion Process

External extraction can be done by either accessing the standard layout file used by the CAD software, or by exporting the layout into an independent representation format. Most packages provide the ability to export their layouts into formats such as STEP, IGES or even Bitmap. Depending on which format is used, the amount of information accessible for extraction, and the specific extraction method will vary.

Regardless of the technicality behind layout data extraction, the CAD drawing should be treated as a database, where relevant information is extracted based on a given set of search criteria. Thus, only the entities that define the model parameters are considered for conversion purposes. In much the same way as the key-concept in database design, a scheme should be employed to mark these entities with a unique set of attributes to simplify the extraction effort. Layer attributes are especially useful for this purpose. For example, a line marking a conveyor can be placed on a layer called CONVEYOR. When the extraction program encounters a line placed on the CONVEYOR layer, it immediately recognizes it as representing a conveyor section.

CAD drawing discipline has to be employed in order to make sure that the respective components will be placed on their designated layers. In the previously mentioned example, only the lines marking a conveyor are allowed on the CONVEYOR layer. This approach introduces the possibility of conversion errors if an entity happens to be placed on the wrong layer. It might therefore be advantageous to use more than one attribute to define a particular element, and always check if all the attributes match in their definitions. Returning once more to the conveyor example, let us assume that the line representing the conveyor always is assigned with the color green. If the

designer mistakenly drops a line representing AGV path on the CONVEYOR layer, but the AGV path is assigned with a different color, his mistake is easily detectable.

Once the information about a model component has been extracted from the layout, it has to be enhanced and converted into a format accepted by the simulation software. This is defined as step B in the conversion process, where the extracted data is translated into a simulation model.

The information obtained from a CAD layout about a model component is usually limited to spatial 2D or 3D coordinate data. If necessary, the drawing can also contain a number of so called descriptive attributes. These can contain value settings and other information that have to be passed to the simulation model and that might change from model to model. Global variables unlikely to change over time should not be incorporated in the layout but should rather be added by the conversion algorithm to the extracted data in step B. This might include general code segments and process subroutines, but also various operational characteristics. Again in the conveyor example, an annotation with the value 10.5 placed in the CONVEYOR layer can indicate the speed of the conveyor segments in this model, as this might vary according to the motor type employed. However, if the width of the conveyor belt always is 2 feet, this information need not be represented in the CAD layout.

The conversion algorithm has to output the model data into a format recognizable to the simulation software. An intermediate file is usually necessary as a medium to which the translated model is written, and which is later read by the simulation software. The flexibility of the intermediate file greatly depends on the simulation software capabilities. AutoMod™ uses the .asy type file format to save its model information. Thus, the output of the conversion algorithm has to mimic an .asy file. Deneb's QUEST™ (Barnes 1997) contains a utility called the Batch Control Language (BCL). BCL is a command language that is used to control QUEST™, and can be thought of as a text-command version of selecting menu in the Graphic User Interface (GUI). BCL command can be used interfacing the model build and/or model run process with data provided by the third party software. By using this approach, the QUEST™ simulation model can be created from a set of text files, describing the coordinate data. In this case, the output of stage B should produce the respective text files.

4 IMPLEMENTATION EXAMPLE

As a matter of business policy, PRI Automation simulates every AMHS layout until its performance is verified to be within customer specifications. Due to the drive to reduce the lead-time of the creation of a semiconductor factory

(Pillai 1999), it is necessary to significantly reduce the design process cycle time. After implementation of cycle time reduction methods (Sokhan-Sanj and Paprotny 1998), the bottleneck in the design process proved to be the creation of the AMHS simulation model. Steps were taken to reduce that task by development of a fully automatic CAD to simulation conversion scheme. The generic conversion process was customized to fit into PRI's general design process, as illustrated in figure 3.

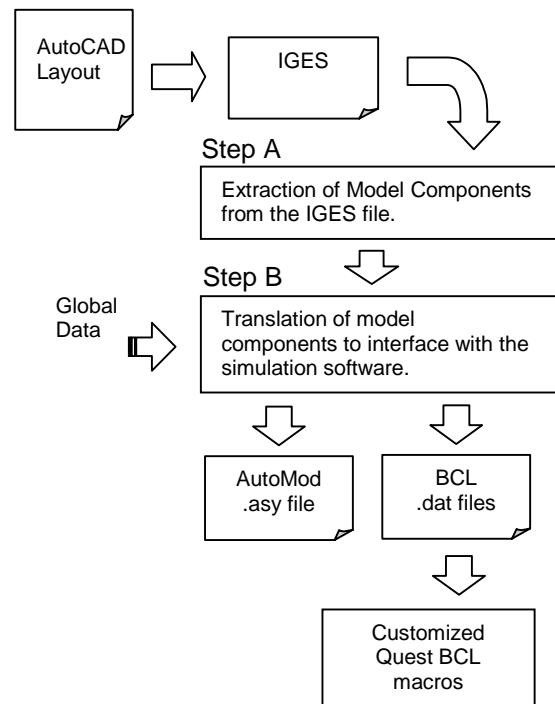


Figure 3: The Customized Version of the Conversion Process

Independent external applications were created for the conversion purpose using the C++ programming language. The format that was chosen as input to step A was an IGES exported CAD layout. IGES stands for Initial Graphics Exchange Specification and is a standard that allows for neutral representation of CAD data. Although several limitations were found (Pasquill 1988), the IGES 80 ASCII characters fixed length format proved quite suitable for data extraction purposes. An intermediate file was added for data storage purpose between step A and B of the conversion methodology. Called the MODEL file, it provides a software independent and portable representation of the AMHS design.

The goal behind the MODEL file was to design a general format that could be used for conversion regardless of the type of simulation software used. This file constitutes the input into the applications of step B.

Depending on whether the target is AutoMod™ or QUEST™, one of the two step B applications is executed. The output is either an .asy file if the model is to be created in AutoMod™, or a number of small .dat files to be processed by the Quest BCL interface. The underlying macros provide a second internal stage of step B within QUEST™, creating the actual simulation model. Figure 4 and 5 illustrate sample models as seen in both simulation packages.

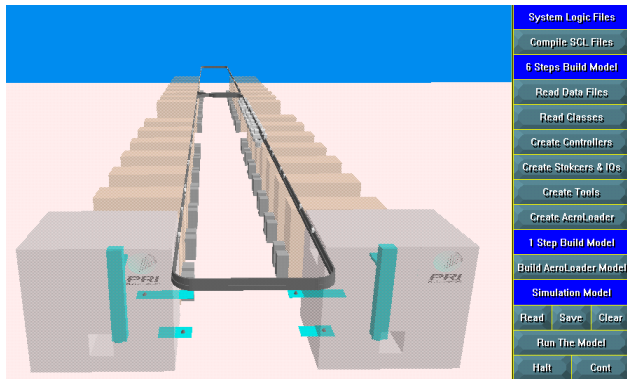


Figure 4: A QUEST PRI Intrabay AMHS Model

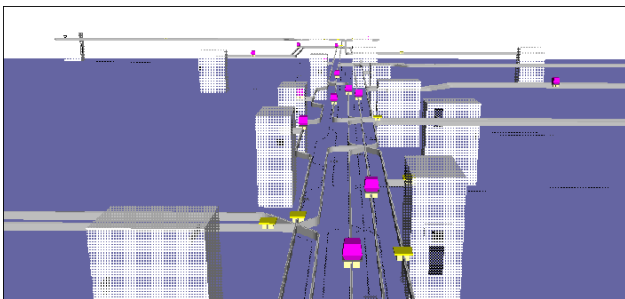


Figure 5: A Typical PRI Interbay AMHS Model Represented in AutoMod

In order to prevent CAD errors from propagating to other stages of the simulation process, extensive error detection algorithm was added to the first part of the conversion process. If the algorithm detects any non-conformities regarding entity attributes or locations in the CAD layout, an error message is placed in a designated IGES file. This IGES file is later imported on top of the CAD layout, and displays the error messages close to the location of the nonconforming entities. This feature assures that a potential CAD error will not propagate to other stages of the simulation process.

5 BENEFITS OF THE IMPLEMENTATION

The implementation of the CAD standards and discipline was based on already employed CAD practices. Thus little or no additional effort was added to the CAD layout creation tasks. However, the amount of layouts needing rework increased due to the error detection algorithm. Prior to implementation of the algorithm, these errors passed undetected through the early stages of the design process, and often had to be fixed on the simulation side. In order to compensate for the increase in CAD effort, additional algorithms were employed to automatically format the AMHS layout for the FactoryFlow™ application. This led to overall reduction in CAD effort required creating a layout design.

The automation of the conversion process has been operational since late summer of 1998. Approximately 40 projects were completed since that time. It is estimated that due to the implementation of automatic CAD to simulation conversion, the model building time was reduced by 90%, and the overall simulation cycle time by 50%. The reduction of simulation cycle time is also attributed to the reduction of error occurrences in the final simulation model, drastically reducing the debugging effort.

Benefits other than effort reduction also became apparent. Automatic conversion of an AMHS layout to a simulation model preserves the dimensional representation of the converted model with its respective layout design. This provides an increase in model validity by eliminating nuisance factors due to differences in applied model building practices. In addition, the ability to quickly and to accurately model the same AMHS design using two distinct simulation software applications shortens model verification efforts and increases flexibility.

6 CONCLUSION

This paper discussed a methodology that significantly reduces the model creation effort for simulations of AMHS designs. Three conditions must be fulfilled in order for the methodology to be applicable. First, a CAD layout of the designed system must exist. Second, the simulation model has to be based on a dimensional accurate representation of the physical system components. Third, the models created must be similar in logic and general structure so that the same CAD transfer template can be re-used. The methodology has to be applicable to a number of subsequent projects in order to justify the implementation effort. If all these conditions hold, automatic model creation will produce significant reductions in the model building effort.

REFERENCES

- Barnes, Martin, R. 1997. An Introduction to QUEST. In *Proceedings of the 1997 Winter Simulations Conference* 5:619-623.
- Pasquill, K.G. 1988. Practical uses of IGES. *Computer-Aided Design* vol. 20, July/August 2:358-359.
- Phillips, Tyler, 1998. AutoMod™ by AutoSimulation. In *Proceedings of the 1998 Winter Simulations Conference*. Vol. 1, 6:213-218.
- Pillai, Dev, 1999. Integrated Factory Design Using Modeling and Simulation. In *Proceedings of the Simulation Solutions'99 Conference*, Institute of Industrial Engineers, 1:7.
- Sokhan-Sanj, Siroos and Paprotny, Igor. 1998. Techniques for Reducing Simulation Model Building Cycle Time. In *Proceedings of the 1998 Summer Computer Simulation Conference* 4:69-72.

AUTHOR BIOGRAPHIES

IGOR PAPROTNY is a Simulation Engineer II in the Automation Planning and Design group at PRI Automation Inc. He joined PRI Automation in 1999. In the past, he has worked on building AMHS simulation models and conducting research in the area of simulation cycle time reduction. He is currently working on projects that aim to streamline the design of AMHS for semiconductor industry. Mr. Paprotny holds a Bachelor Degree in Industrial Engineering from Arizona State University and an Engineering Diploma in Mechatronics from the NKI College of Engineering, Oslo.

GERALD T. MACKULAK is an Associate Professor in the Department of Industrial Engineering at Arizona State University. His research interests center on improvements to simulation methodology, material handling modeling, and production scheduling in high technology manufacturing. He holds B.S.I.E., M.S.I.E. and Ph.D. degrees from Purdue University.

WENDY ZHAO is Senior Simulation Engineer in the software division at PRI Automation. She has developed AMHS simulators with a great deal of accuracy, flexibility and speed of modeling for end users in QUEST. She has worked on many simulation projects to evaluate the performance of the existing and proposed AMHS. Prior PRI Ms. Wendy Zhao worked as an industrial engineer.