

Waldemar Małopolski  orcid.org/0000-0001-6927-3320

malopolski@mech.pk.edu.pl

Production Engineering Institute, Faculty of Mechanical Engineering,
Cracow University of Technology

A METHOD FOR THE AUTOMATIC CREATION OF BIDIRECTIONAL TRANSPORTATION PATHS FOR AGVs IN DELMIA QUEST

METODA AUTOMATYCZNEGO GENEROWANIA DWUKIERUNKOWYCH DRÓG TRANSPORTOWYCH DLA ROBOTÓW MOBILNYCH W PROGRAMIE DELMIA QUEST

Abstract

In this paper, a method for the automatic creation of bidirectional paths for AGVs in the Delmia QUEST software application is presented. The layout of transportation subsystem can be prepared in a spreadsheet. Based on it the file with input data is generated. Using programming languages included in QUEST, procedures were created and based on these procedures, a macro was built. This macro allows reading input data from the file and creates the transportation paths automatically. This enables the simulation model building to be less time consuming.

Keywords: modelling, simulation, bidirectional paths, AGV, Delmia QUEST

Streszczenie

W artykule przedstawiono metodę automatycznego generowania dwukierunkowych dróg transportowych w programie Delmia QUEST dla autonomicznie sterowanych robotów mobilnych. Układ i rozmieszczenie dróg transportowych są projektowane w arkuszu kalkulacyjnym, a następnie zapisane jako dane wejściowe w postaci pliku tekstowego. Wykorzystując języki programowania zawarte w programie QUEST, opracowano odpowiednie procedury. Zostały one następnie wykorzystane do budowy macro, które wczytuje dane z pliku i w automatyczny sposób generuje całą sieć dróg transportowych. Dzięki temu budowanie modelu symulacyjnego może być uproszczone i przyspieszone.

Słowa kluczowe: modelowanie, symulacja, dwukierunkowe drogi, roboty mobilne, Delmia QUEST

1. Introduction

Transportation subsystems currently play a very important role in production systems. They are used to increase the flexibility and decrease the production costs. Nowadays, transportation subsystems are very often based on automated guided vehicles (AGVs). This kind of solution guarantees a high flexibility and efficiency. However, there are many problems associated with the AGVs construction and control; for example, some aspects of design and operational parameters of AGVs are presented in [1]. The biggest problem is to develop collision-free and deadlock-free methods of controlling AGVs. In this area, various research studies are still being conducted. For example, one method of handling deadlocks in automated manufacturing systems based on AGVs is described in [2] and an AGV scheduling algorithm and collision avoidance is presented in [3]. A cyclic scheduling problem with AGVs is presented in [4]. The concept of a distributed AGV control system is presented in [5]. A collision and deadlock free method for decentralised route planning in a dynamic environment is described in [6]. The assignment of suboptimal AGV routing and scheduling is considered in [7]. A novel paradigm for conflict resolution in multi-vehicle traffic systems is presented in [8]. A new approach to dynamic control of the motion of multiple mobile robots can be found in [9]. A genetics-based machine learning approach for collision free AGV control is presented in [10].

Simulation programs, especially commercial programs, are very often used in the design of production or transportation systems, including those with AGVs. In this area, 3D modelling and simulation capabilities are becoming increasingly available. One of the most popular software programs with full 3D functionality is Delmia QUEST. These types of simulation tools are used, for example, for the analysis of vehicle control and functionality together [11]. An approach for job shop scheduling based on a simulation approach is presented in [12]. The application of virtual reality in order to present a methodology for the requirement analysis of holonic manufacturing system design is shown in [13]. It is also possible to simulate the operation of a production line and estimate the number of pallets in workshops [14]. We can find many different applications of simulation to analyse the operation of production systems and transportation subsystems.

Using such powerful tools for the manual modelling of large and complex systems is very time consuming. Therefore, we can find different solutions to facilitate and speed up modelling, for example, a method of creating a simulation model in which production information is transferred through the extensible mark-up language (XML) is presented in [15], but this solution is dedicated to aircraft assembly production line. Other XML-based input data specifications are used for the creation of models and for executing simulations [16]. In [17], a tool for rapid layout description and a 3D discrete-event model creation are proposed. In [18], the interface between manufacturing software applications including process planning systems and discrete-event simulation tools was developed. In another solution, a simulation model based on an appropriate input and output data is generated. This model is used for analysis of the sustainability of the manufacturing system [19].

Using the above-described methods and solutions it is possible to reduce both model-preparation time and the cost of simulation experiments; nevertheless, creating very powerful

and flexible supporting tools is particularly difficult. With the increase of their universality, they become very complicated. Moreover, these tools are usually dedicated for specific applications.

Another possible solution is to build a tool that will enable the automatic performance of simple but repetitive and time-consuming activities. This type of activities we do during AGV paths creation. We have to repeat these activities many times, especially if the path network is very large. Therefore, in this paper a method for the automatic creation of bidirectional transportation paths for AGVs is proposed. This method was developed for the Delmia QUEST program.

2. Simple path description

As described above, the proposed method should be easy and simple. Furthermore, the process of preparing input data should be easy to implement. For this reason, the division of the entire transportation area into squares is proposed. In this example, the sides of every square are set to 1 m. Under this assumption, the entire transportation network can be defined in a spreadsheet (Fig. 1).

E	X	X	O	O	O	E	O	O	O	E	O	O	O	O	O	E	O	O	E	O	O	E	O	E	O	E				
O	O	X	X	O	O	X	O	C	O	X	O	O	O	O	O	X	O	O	X	O	O	X	O	X	O	X				
O	O	O	X	X	X	X	X	X	X	O	O	O	O	O	O	X	X	X	X	O	O	X	X	X	X	X				
E	X	O	X	O	X	O	O	X	O	O	O	O	X	X	X	X	O	X	O	O	O	O	X	X	X	O	O	X	O	
O	X	X	X	O	X	O	O	X	X	X	X	X	O	E	O	O	X	X	X	O	O	S	O	O	O	O	X	O		
O	O	X	O	O	S	X	X	X	O	O	O	O	S	O	O	O	O	O	O	X	X	X	X	O	O	X	X	X	O	
O	O	X	X	X	X	O	X	O	O	O	O	O	X	X	X	X	O	O	X	O	O	O	O	O	X	O	C	O		
O	O	X	O	O	O	O	X	X	X	X	O	O	X	O	O	O	X	O	O	X	X	X	O	C	O	X	O	O	O	
E	X	X	X	O	O	O	O	X	O	X	O	X	X	O	O	O	X	E	O	O	O	X	X	X	X	X	O	O	O	
O	O	O	X	O	O	O	O	S	O	X	O	X	O	O	O	O	X	O	O	O	O	S	O	O	O	X	X	X	E	
O	O	O	X	C	O	O	O	X	O	X	O	X	C	O	X	X	X	X	O	O	O	X	O	O	O	O	X	O	O	
O	O	O	X	O	O	O	O	X	X	X	O	X	O	O	X	O	O	X	O	X	X	X	O	O	O	O	X	O	O	
O	O	O	X	O	O	S	X	X	O	X	O	X	X	O	O	X	O	O	X	O	X	O	X	X	X	X	O	O	O	
E	X	X	X	O	O	X	O	O	O	X	O	O	X	O	X	X	S	X	O	X	O	X	O	X	O	X	O	O	O	
O	O	O	X	X	X	X	C	O	O	X	O	E	X	X	X	O	X	O	O	O	O	X	X	S	O	X	O	O	O	
O	O	X	X	O	O	X	O	O	X	X	O	O	X	O	O	X	O	O	O	X	O	O	O	X	O	O	X	O	O	
E	X	X	O	O	O	E	X	O	O	X	O	O	O	S	X	X	X	O	O	S	X	X	X	X	O	X	X	X	E	
O	X	O	O	O	O	S	O	O	X	X	X	X	X	X	O	X	O	O	O	X	O	O	O	X	O	O	X	O	O	
O	X	X	X	X	X	X	O	O	O	O	O	X	O	O	O	X	X	X	X	O	O	O	X	X	X	X	O	O	O	
E	X	O	O	O	O	E	O	O	O	O	O	E	O	O	O	E	O	O	O	O	O	O	O	O	O	O	O	X	X	E

Fig. 1. The layout of a transportation subsystem in spreadsheet

A single cell corresponds to a 1 x 1 m square. In this example, the number of columns is 30 and the number of rows is 20. Bidirectional roads are marked with 'X' on a green background. Impassable places are marked 'O' on a white background. Parking places for the

AGVs are marked by 'C' on a yellow background. Places on the roads where AGVs can be loaded or unloaded are marked 'S' on a red background. Places where work pieces may be introduced in or out of the transportation subsystem are marked 'E' on a blue background. In this way, a large and complex transportation subsystem can be quickly designed. Projects of transportation layout prepared in this way are saved in the *.csv file format. The part of this file is shown in Fig. 2a. In this file, all signs are separated by semicolon. In the next step, semicolons have to be converted to space signs. Each sign in this file, besides spaces, corresponds to one square. The part of this file with input data is presented in Fig. 2b.

```
E;X;X;O;O;O;E;O;O;O;E;O;O;O;O;O;E;O;O;E;O;O;E;O;O;E;O;E;O;E
O;O;X;X;O;O;X;O;C;O;X;O;O;O;O;O;X;O;O;X;O;O;X;O;O;X;O;X;O;X
O;O;O;X;X;X;X;X;X;X;X;O;O;O;O;O;X;X;X;X;O;O;X;X;O;X;X;X;X;X
E:X:O:X:O:X:O:O:X:O:O:O:O:X:X:X:X:O:X:O:O:O:O:X:X:X:O:O:X:O
```

(a) A part of input data in *.csv file format

```
E X X O O O E O O O E O O O O O E O O E O O E O E O E
O O X X O O X O C O X O O O O O X O O X O O X O O X O X O X
O O O X X X X X X X X O O O O O X X X X O O X X O X X X X X
E X O X O X O O X O O O O X X X X O X O O O O X X X O O X O
```

(b) A part of input data in *.dat file format

Fig. 2. The parts of input data files

The input data in this format can be used for the automatic creation of paths. Any two adjacent squares on the road should be replaced by a single segment in the simulation model. The start point and the end point of each segment should be placed in the centre of adjacent squares.

3. Internal programming languages in QUEST

QUEST is a very popular and advanced tool for discrete-event modelling and simulation; it has developed functionality for 3D model creation. This functionality includes the ability to build models of transportation subsystems based on conveyors or AGVs. Each kind of model can be built using a graphical user interface. In the case of building simple models of transportation subsystems, manual data entry is quite effective. However, when we have to build a large and very complex model, as in this case, the approach described above is ineffective. The process of creating such a complex model can be facilitated through the use of programming languages which are built-in QUEST.

3.1. Simulation control language

The simulation control language (SCL) is a high level procedural language. This enables the construction of logic to control the actions and behaviour of each model object. SCL logic can be used to control the following:



- ▶ routing,
- ▶ processing,
- ▶ queueing,
- ▶ AGV/labour motion,
- ▶ decision point activity,
- ▶ initialisation, termination, simulation,
- ▶ pre- and post-event actions,
- ▶ the behaviour of user defined SCL buttons/macros.

In addition to the applications mentioned above, SCL logic can be used to define several data types, procedures and routines. In this way, the programmer can access different information about each object of the model. SCL provides many standard input and output routines to streams like files, pipes and sockets. Using SCL, the programmer can write procedures for input data file reading.

3.2. Batch control language

The batch control language (BCL) is a command language that is used to control QUEST. BCL commands can be used, inter alia, for the following tasks:

- ▶ reading a model,
- ▶ modifying parameters,
- ▶ running a simulation,
- ▶ interrogating the simulation results,
- ▶ providing visual control.

By using BCL commands, we can provide data to QUEST. What is more important is that it is possible to set model parameters from SCL by calling BCL commands. It is possible to use SCL and BCL language simultaneously; this makes it possible to build procedures for the automatic creation of a model based on input data from a file.

4. Automation of path creation

In the first step, input data was saved in a file named MapOfNetwork.dat. The SCL routines were used to read input data from this file. All this data was stored in the table named MapOfNetwork. A part of the procedure in the SCL language for file reading is shown below.

The part of procedure in SCL

```
open file '../MapOfNetwork.dat' for text input as 1
  for x = 0 to (R_dimension-1) by 1 do
    for y = 0 to (C_dimension-1) by 1 do
      read( #1, MapOfNetwork[x,y])
    endfor
  endfor
close #1
```

In the next step, all segments of AGV paths were created based on this data. All segments joined together then had to be logically connected. In this case, there was a problem in the SCL. The routine which logical connects segments after performing an operation changes the position and orientation of the segments (Fig. 3). Two segments are shown in Fig. 3 – before logical connection (Fig. 3a) and after connection (Fig. 3b).

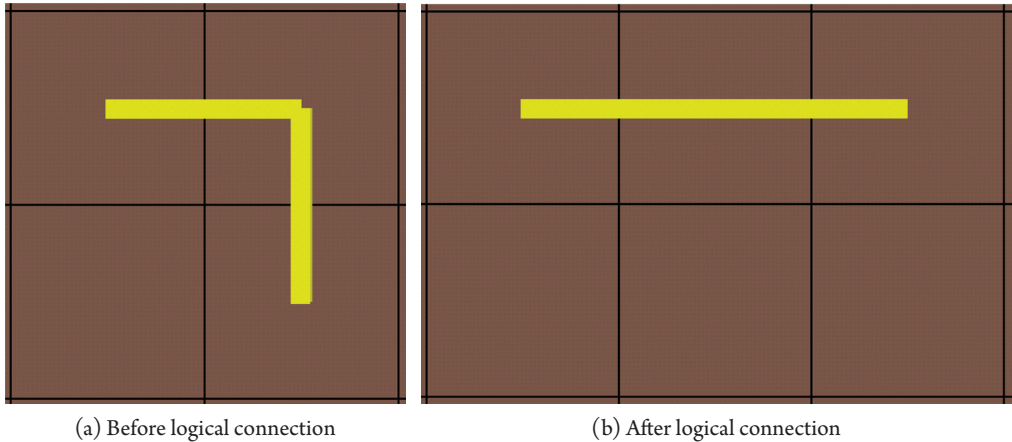


Fig. 3. Two segments of road

To solve this problem, a different approach was applied. In the first step, all segments were created in the same starting point. In this case, a BCL command was called from SCL. The part of the SCL procedure which was used for segment creation is shown below.

The part of procedure in SCL

```

...
for x = 0 to (R_dimension-1) by 1 do
  for y = 0 to (C_dimension-1) by 1 do
    if (MapOfNetwork[x,y]<>'O') then
      if (y < C_dimension-1) then
        if (MapOfNetwork[x,y+1]<>'O') then
          if ((MapOfNetwork[x,y+1]=='C') OR
              (MapOfNetwork[x,y+1]=='E')) then
            bcl_status = BCL("CREATE LINEAR SEGMENT FOR
              'AGV_Path_Network' FROM COORD -14.5,9.5,0 TO -13,9.5,0")
          ...
        endifor
      ...
    endifor
  ...

```

Following the part of the procedure presented above, appropriate segments were logically connected and again moved to the starting point. The part of the SCL procedure which was used for the connection of segments is shown below.

The part of procedure in SCL

```
...
for y = 0 to (C_dimension-1) by 1 do
  if (MapOfNetwork[x,y]<>'O') then
    ...
    Bcl_Str = "CONNECT SEGMENT
              'AGV_Path_Network_1."+Y1_Str+"
              FORWARD OUTPUT TO SEGMENT
              'AGV_Path_Network_1."+X1_Str+"
              REVERSE INPUT"
    bcl_status = BCL(Bcl_Str)
    Bcl_Str = "TRANSLATE SEGMENT
              'AGV_Path_Network_1."+X1_Str+" TO -14.5,9.5,0"
    bcl_status = BCL(Bcl_Str)

    Bcl_Str = "TRANSLATE SEGMENT
              'AGV_Path_Network_1."+Y1_Str+" TO -14.5,9.5,0"
    bcl_status = BCL(Bcl_Str)
  ...
```

Finally, all segments were moved to the destination places and were rotated if necessary. The part of the SCL procedure which was used for the movement and rotation of segments is shown below.

The part of procedure in SCL

```
...
for x = 0 to (R_dimension-1) by 1 do
  for y = 0 to (C_dimension-1) by 1 do
    ...
    counter = counter + 1
    if ((MapOfNetwork[x,y]=='C') OR (MapOfNetwork[x,y]=='E'))
    then y1 = 10 - x
    else y1 = 9.5 - x
    endif
    x1 = y - 14.5
    Count_Str = str('%0f', counter)
    X1_Str = str('%1f,', x1)
    Y1_Str = str('%1f,', y1)

    Bcl_Str = "TRANSLATE SEGMENT
              'AGV_Path_Network_1."+Count_Str+"
              TO "+X1_Str+Y1_Str+"0"
    bcl_status = BCL(Bcl_Str)
    Bcl_Str = "ROTATE SEGMENT
```



```

'AGV_Path_Network_1."+Count_Str+"'
TO 0,0,-90"
bcl_status = BCL(Bcl_Str)
...

```

An example of the connected segments is presented in Fig. 4. A view of all transportation subsystems is presented in Fig. 5; this corresponds to the layout designed in the spreadsheet shown in Fig. 1. In this way, all segments were appropriately created and connected.

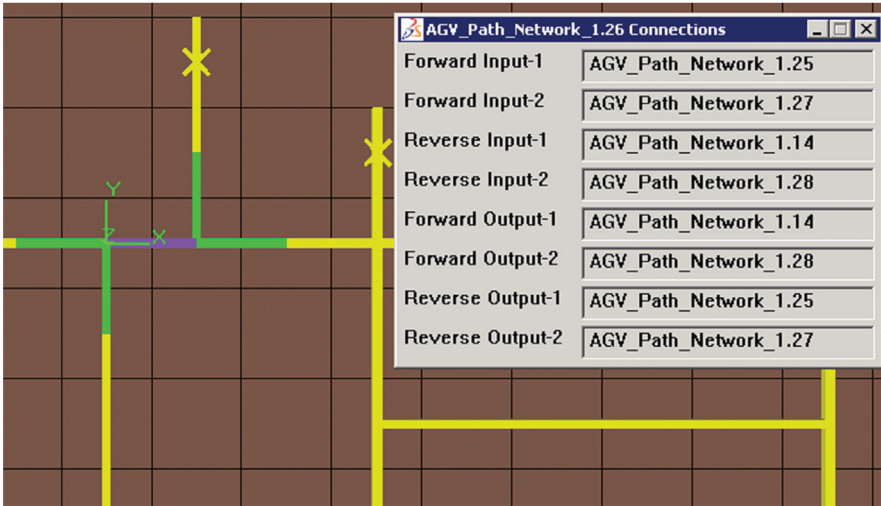


Fig. 4. An example of segment connections

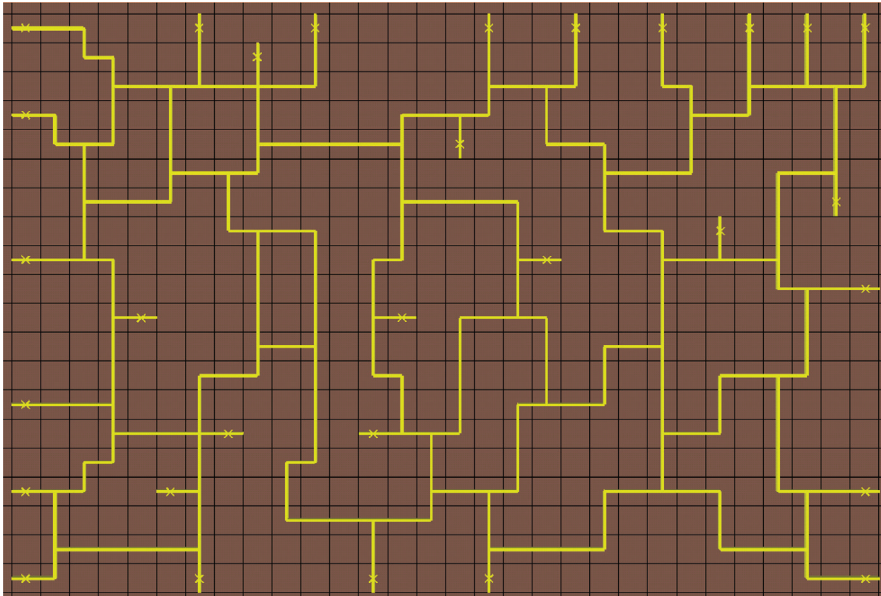


Fig. 5. All segments of a transportation subsystem created in QUEST

5. Final result

Procedures developed in SCL and BCL were implemented into a macro in QUEST. This enables the automatic creation of transportation paths according to the layout designed in a spreadsheet. In the presented example, there are more than 270 segments that have been created and appropriately connected. Doing this manually would be very time consuming. Moreover, it would be very easy to make a mistake.

After all paths creation additional objects were added to the model manually, among others one AGV. Finally, transport tasks were simulated across the transportation paths. In this way, the operation of the transportation subsystem has been tested (Fig. 6).

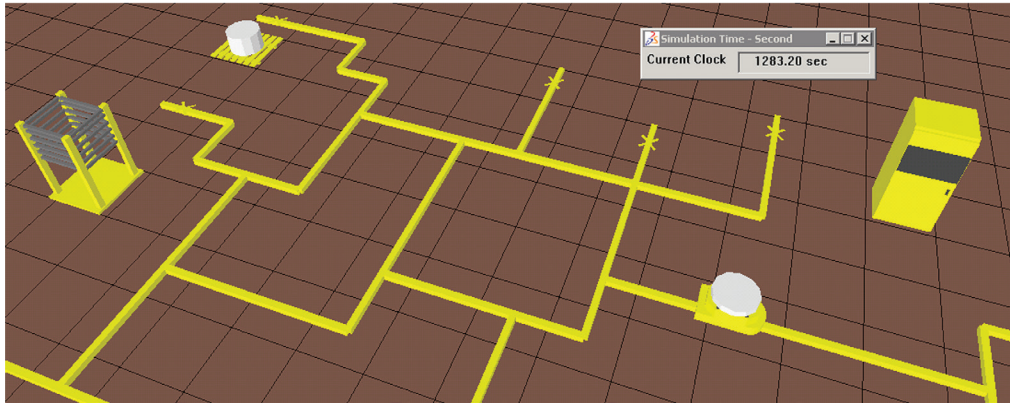


Fig. 6. A 3D view of the simulation process in Delmia QUEST

6. Conclusion

The proposed method is very easy and fast to implement using SCL and BCL in QUEST. The main goal of this approach is to speed up the creation of models. This method is focused on automating repetitive operations and thus avoids a loss of time for simple tasks associated with model building. Using a spreadsheet, we can prepare many different transportation layouts. What is important is that we do not need any specialised tools to achieve this; therefore, we can concentrate on model development or the implementation of a new AGV control strategy. This method can also be applied to build models in other simulation tools, for example, in Arena, based on transportation network defined in a spreadsheet, it can be achieved using Visual Basic.

References

- [1] Zajac J., Chwajol G., Więk T., Krupa K., Malopolski W., Słota A., *Automated Guided Vehicle System for Work-in-Process Movement*, Solid State Phenom., Vol. 196, 2013, pp. 181–188.
- [2] Zajac J., *A deadlock handling method for automated manufacturing systems*, CIRP Ann. Manuf. Technol., Vol. 53, Issue 1, 2004, pp. 367–370.
- [3] Akturk M.S., Yilmaz H., *Scheduling of automated guided vehicles in a decision making hierarchy*, Int. J. Prod. Res., Vol. 34, No. 2, 1996, pp. 577–591.
- [4] Bocewicz G., Banaszak Z., *Declarative approach to cyclic steady state space refinement: periodic process scheduling*, Int. J. Adv. Manuf. Tech., Vol. 67, 2013, pp. 137–155.
- [5] Liu S., Gruver W., Kotak D., Bardi S., *Holonic manufacturing system for distributed control of automated guided vehicles*, 2000 IEEE International Conference on Systems, Man and Cybernetics, 2000, pp. 1727–1732.
- [6] Ando M., Nishi T., Konishi M., Imai J., *Autonomous decentralized route planning method and its application to a dynamic route planning*, 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation, 2003, pp. 718–723.
- [7] Bocewicz G., Nielsen I., Banaszak Z., *Automated guided vehicles fleet match-up scheduling with production flow constraints*, Eng Appl Artif Intell., Vol. 30, 2014, pp. 49–62.
- [8] Reveliotis S., Roszkowska E., *Conflict resolution in multi-vehicle systems: A resource allocation paradigm*, 4th IEEE Conference on Automation Science and Engineering, 2008, pp. 115–121.
- [9] Roszkowska E., *Provably correct closed-loop control for multiple mobile robot systems*, Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2015, pp. 2810–2815.
- [10] Sakakibara K., Fukui Y., Nishikawa I., *Genetics-Based Machine Learning Approach for Rule Acquisition in an AGV Transportation System*, 2008 Eighth International Conference on Intelligent Systems Design and Applications, Vol. 3, 2008, pp. 115–120.
- [11] Moore P., Ujvari S., Pu J., Lundgren J.-O., Xie C., *Intelligent semi-autonomous vehicles in materials handling*, Mechatronics, Vol. 9, 1999, pp. 881–892.
- [12] Wang C., Qiu C., *Virtual simulation of the job shop scheduling system based on delmia/QUEST*, 7th Intl. Conf. on Sys. Simulation and Scientific Computing, 2008, pp. 1129–1132.
- [13] Manesh H.F., Schaefer D., Hashemipour M., *Information requirements analysis for holonic manufacturing systems in a virtual environment*, Int. J. Adv. Manuf. Tech., Vol. 53, 2011, pp. 385–398.
- [14] Guan Z., Cao L., Wang Ch.J., Cui Y., Shao X., *Simulation of Logistics System with Aspect of Pallet Requirements Optimization Based on Digital Factory*, Advances in Automation and Robotics, Vol. 1, 2011 International Conference on Automation and Robotics (ICAR 2011), 2011, pp. 293–302.
- [15] Du J., He Q., Fan X., *Automating generation of the assembly line models in aircraft manufacturing simulation*, 2013 IEEE International Symposium on Assembly and Manufacturing (ISAM), 2013, pp. 155–159.

- [16] Lu R.F., Qiao G., Riddick F.H., McLean C., *NIST XML simulation interface specification at Boeing: a case study*, 2003 International Conference on Machine Learning and Cybernetics, Vol. 2, 2003, pp. 1230–1237.
- [17] Han K.H., Bae S.M., Choi S.H., Lee G., Jeong D.M., *Parametric layout design and simulation of flexible manufacturing system*, Recent Researches in Power Systems and Systems Science, 2011, pp. 94–99.
- [18] Fournier J., *Model building with core manufacturing simulation data*, 2011 Winter Simulation Conference, 2011, pp. 2219–2227.
- [19] Lee J., Kang H.S., Noh S.D., *Simulation-based analysis for sustainability of manufacturing system*, Int. J. Pr. Eng. Man-Gt., Vol. 13, No. 7, 2012, pp. 1221–1230.

