# Other Applications of Multimedia and Signal Processing - Volume 7

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A New Fast Hybrid Algorithm of Background Extraction Suitable for the DSP Platform</td>
<td>1</td>
</tr>
<tr>
<td>Li Xiaofei</td>
<td></td>
</tr>
<tr>
<td>Regional Style Automatic Identification for Chinese Folk Songs</td>
<td>5</td>
</tr>
<tr>
<td>Yi Liu, Lei Wei, and Peng Wang</td>
<td></td>
</tr>
<tr>
<td>Determination of Total Amino Acids in Oilseed Rape Leaves Using Near Infrared Spectroscopy and Chemometrics</td>
<td>10</td>
</tr>
<tr>
<td>Fei Liu, Fan Zhang, Hui Feng, Weijun Zhou, and Yong He</td>
<td></td>
</tr>
<tr>
<td>A 3D Display System for Cloth Online Virtual Fitting Room</td>
<td>14</td>
</tr>
<tr>
<td>Xiao Hu Liu and Yu Wen Wu</td>
<td></td>
</tr>
<tr>
<td>Spatial-Temporal Distortion Metrics for Video</td>
<td>19</td>
</tr>
<tr>
<td>Chunying Yang, Lin Zhao, and Zatyi Liao</td>
<td></td>
</tr>
<tr>
<td>Abnormal Noise Detection Method Based on Wavelet Filter and K-L Information</td>
<td>24</td>
</tr>
<tr>
<td>Zhang Gen-Yuen</td>
<td></td>
</tr>
<tr>
<td>Multimedia System Design Illustrating the Operation of Doppler Ultrasound Fetal Monitor in Extracting Cardiac Information</td>
<td>30</td>
</tr>
<tr>
<td>N.A. Md Norani and W. Mansor</td>
<td></td>
</tr>
</tbody>
</table>
MDD Based CASE Tool for Modeling Heterogeneous Multi-jointed Robots .................................................. 775
Woo Yeol Kim, Hyun Seung Son, R. Young Chui Kim, and C.R. Carlson
Production Design by Simulation Software “Witness” – A Case Study ...................................................... 780
MD Sarder and Sumanth Yenduri
From Craft to Engineering; Software Development and Schema Theory ................................................ 787
Mark Burgin

Theory and Formal Methods - Volume 7

Distributed Viewpoint Equivalences for Higher Order Processes ...................................................... 792
Zining Cao
Convention-Based Syntactic Descriptions ......................................................................................... 797
Ray Toal and Derek Smith
Formalism for Safe Component Upgrades ....................................................................................... 802
Xiaoshui Xu, Linpeng Huang, and Xin Wang
Reliability Analysis in the Early Development of Real-Time Reactive Systems ................................ 807
Miao Zheng and Olga Ormanajieva
Formal Modeling and Analyzing Kerberos Protocol ........................................................................ 813
Qin Li, Fan Yang, Huibao Zhu, and Longjie Zhu
Execution Semantics Analysis Based Composition Compensation Mechanism in Web Services Composition ................................................................. 820
Xiaoyong Mei, Aijun Jiang, Fuden Zheng, and Shixin Li
Modeling Nondeterministic Feature with Petri Net for Network Protocol in Interoperability Testing ................................................................. 825
Li Hua, Ye Xinning, Wu Chengyong, Hang Chenghao, and Wang Lingling
Structuring Cognitive Information for Software Complexity Measurement ........................................ 830
Benjapo Supraseri and Yatchai Limpyakorn
The Formal Model of Real-Time Service Components Composition Based on DCValid ....................... 835
Xianli Jin and Xi Shao
Automatic Generation Method of Assembly Tolerance in Large-Scale Assembly Design ...................................... 840
Yi Zhang, Zongbin Li, and Liping Zhao
An Innovative Bucket Sorting Algorithm Based on Probability Distribution ...................................... 846
Zhongxiao Zhao and Chen Min

Author Index - Volume 7 .......................................................................................................................... 851
MDD based CASE Tool for Modeling Heterogeneous Multi-Jointed Robots

Woo Yeol Kim¹, Hyun Seung Son¹, R. Young Chul Kim¹,², C. R. Carlson²
¹Dept. of Computer & Information Comm., Hongik University, Korea
{john, son, boh}@selab.hongik.ac.kr¹
²Dept. of ITM, Illinois Institute of Technology, Wheaton, IL USA

Abstract
In this paper, we develop MDD based case tool for modeling heterogeneous multi-jointed robots. This automatic tool consists of the whole procedure from design to implementation based on our extended UML, which automatically generates heterogeneous models. We also add pre-simulation step in this tool for making an accurate estimate of multi-jointed robot’s motions before developing it.

Keywords: Model Driven Development, CASE Tool, Multi-Jointed Robot, Unified Modeling Language, Modeling.

1. Introduction
Recently, multi-jointed robots are applied in many fields such as risk construction area¹, military equipment movement², disaster zone³. The more the human increasingly use these multi-jointed ones in many fields, the more this embedded Software would be very complex. Also we need to cooperate the related stakeholders and to choose the better technique of software designs for quickly developing the complicated software with reliability⁴. Due to this reason, there is growing tendency to develop the system with UML⁵ mechanism. The previous UML tools are Rhapsody⁶, TAU G2⁷, Rational RT⁸, and so on. These tools have a lot of functions, but cannot be customized to develop some embedded software systems. So, they just play a role to design on most cases.

In this paper, we propose to solve the previous UML design tool problems based on MDD [⁹,¹⁰,¹¹] tool, and in addition to mention ‘modeling & simulation’ for correctly moving the complex multi-jointed robot.

Our proposed tool also uses to model the system with use case diagram, class diagram, concurrent message diagram, and concurrent state diagram based on MDD (Model Driven Development) that consists of TIM(Target independent Model), TSM(Target Specific Model) and TDC(Target Dependent Code) [⁹,¹⁰,¹¹]. Using profile mechanism, it can automatically transform TSM and then generate Code. To reliably and correctly make heterogeneous software, it should be necessary to make a whole automatic process of our tool.

This paper is organized as follows: we introduce MDD(Model Driven Development) in section 2. In section 3, it shows our proposed development process. In section 4, it shows to design heterogeneous multi-jointed robot with our tool, and analyzes/compare with the existing tools. Finally we mention a conclusion.

2. Related Works
In robot fields, most researches focus on developing embedded software, but it is very hard to reuse embedded software on heterogeneous embedded systems due to hardware oriented development, which is just code-oriented development. MDA(Model Driven Architecture) mechanism is based on one meta-model describing the systems to be built. A system description is made of numerous models, each model representing a different level of abstraction. The modeled system can be deployed on one or more platform via model to model transformations.

MDD automatically transforms design/model into the executable system, and reuses the iteratively redefined model, which is gradually software development process. In that time, model transformation executes automatically in partially/whole procedure.

3. Proposed Development Process
In this section, we mention MDD based development process and model transformation algorithm.
3.1 MDD based Development Process

Our proposed development process in figure 1 consists of five activities of requirement analysis, M&S(Modeling&Simulation), TIM(Target Independent Model), TSM(Target Specific Model), TDC(Target Dependent Code: that is, code generation). First, in requirement analysis step, we can analyze a general target system with use case diagram. Second, in M&S step, we can assemble into a virtual target system, and also simulate to correctly move the virtual target robot. Third, in TIM step, we can design the independent system on middleware(s), operating system(s), and processor(s). We also model the static view of the general system with class diagram, and the dynamic view with concurrent message diagram and concurrent state diagram. Fourth, in TSM step, we can convert the general target independent model into several target specific models which are satisfied on the particular system related with middleware, OS and processor. Finally, in TDC step, it is the code generation step of each specific target model. We can automatically generate code(such as C, C++ and Java) through each specific model as the automatic code generation. Our proposed modeling tool which is based on MDD based development process supports to automatically transform model-to-model and to generate source code.

3.2 Model Transformation Algorithm

This core mechanism for modeling heterogeneous multi-jointed robots should make the transformation algorithm from a general target model to several targets such as robot1 and robot2 in figure 1. We may call 'this model transformation algorithm' into ‘model compiler’. Figure 2 shows to flow the whole algorithm. TIM step will generate semantic tree and context table by analyzing the scanning process of diagrams, that is, meta-models of class diagrams, concurrent message diagrams and concurrent state diagrams. The semantic tree has the values of notations in each diagram. The context table means to generate the standardized meta-model table. After the above process, it creates profile table through loading profile, and also middle language to map the context table with profile table. In this case, it should keep a rule for mapping one-to-one. The middle language generates the TSM(s) applied with model transformation rule from a TIM.

\[ \text{Robot Model Data} \rightarrow \text{Robot Behavior Data} \rightarrow \text{Simulation} \rightarrow \text{TIM} \rightarrow \text{Diagram Scanning} \rightarrow \text{Semantic Tree Creation} \rightarrow \text{Context Table Creation} \rightarrow \text{Transformation Rule Load} \rightarrow \text{Transformation Rule Analysis} \rightarrow \text{Middle Language Creation} \rightarrow \text{TSM} \]

Figure 2. Model transformation algorithm

4. MDD based Modeling Tool

4.1 Proposed modeling tool structure

The modeling tool classifies with UML model, profile parser, model transformer and model simulator in figure 3. Use case diagram, class diagram, concurrent message diagram and concurrent state diagram are made to use in UML model. To do this attempts to reduce the redundant notations of each diagram. It may have one advantage for easily creating diagram in UML model when we need a new diagram.
for new functions. The profile parser can be one interpreting module to read profiles of middleware(s), OS(s) and processor(s), and adopt to transform model together. The model transformer can transform one general TIM into several TSMs. The model simulator can simulate to correctly move the virtual target object.

shows to compose the Robot with the model assembler.

![Figure 3. The whole structure of CASE tool](image)

**4.2 Modeling for Multi-Jointed Robot**

![Figure 4. Model assembler for multi-jointed robot](image)

The whole developing process for the robot does execute this order of M&S(Modeling & Simulation), TIM, TSM and TDC. In this time, we use the robot model assembler and simulator at M&S step, and apply MDD based modeling tool at each TIM, TSM and TDC step. The robot model assembler can assemble parts of the robot to generate the virtual robot. So, we can possibly make various types of the virtual robot according to combing parts in other orders. Figure 4 shows to compose the Robot with the model assembler.

![Figure 5. Simulator for multi-jointed robot](image)

The simulator in the virtual environment is a tool of simulating with the motion data of the virtual robot developed by the model assembler.

We can test the robot motion like the real world within this simulator which is applied by the physical engine. We can also use dashboard to control the virtual robot in the virtual environment. Figure 5 is the model simulator for multi-jointed robot, which may find some robot’s mistakes before completely developing it.

Figure 6, 7, 8 shows the modeling steps for heterogeneous robots with our development tool.

![Figure 6. TIM design of heterogeneous multi-jointed robot](image)

First, Figure 6 shows the TIM Design of the robots. We can see two dotted rectangles, that is, 'Ultrasensor' class and 'ServoMotor' class, but each dotted
rectangle, ultra sensor, is different in Figure 7, 8. In other words, we use two different ultra sensors sensing to move forward/backward/left-turn/right-turn.

'MultiJointedRobot' class controls the robot behaviors with receiving data via sensor. 'RobotMotion' class brings the stored motion data to execute the determined robot behaviors. 'Arm', 'Joint' and 'ServoMotor' are also classes to perform the robot motions. 'UltraSensor' class handles data of ultra sensor.

![Figure 7. TSM design of ROBOT 1](image)

Like Figure 7, Robot is generated 'DG-118S' class as servo motor, and 'NT_TS601' class as ultra sensor. The dotted class is dependent on robot hardware. That is, we choose DG_118S an NT_TS601 in our tool, then automatically generate then like Figure 7.

Figure 8 shows to transform the TIM (in Figure 6) into other TSM. We use the same servo motor like Robot1, but the different ultra sensor (SRF04) like Robot2.

![Figure 8. TSM design of ROBOT 2](image)

4.3 The Comparison of UML Tools

To analyze the advantages of tools, we compare the existing tools with the particular attributes. These tools except TAU G2 are focused on embedded systems. TAU G2 is focused on the Enterprise application, and not supported with real-time, simulation and parallel mechanism. It is very important that we should choose the right tool to develop the particular target even if these are the existing tools with many functions.

<table>
<thead>
<tr>
<th>Target</th>
<th>Rhapsody</th>
<th>TAU G2</th>
<th>Rations 1 RT</th>
<th>Our proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>MED/MDA support</td>
<td>Embedde</td>
<td>Embedde</td>
<td>Embedde</td>
<td>Embedde</td>
</tr>
<tr>
<td>Heterogeneous platform</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>Real Time Design</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Code generation</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>Simulation support</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Heterogeneous E/W support</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>Parallel mechanism</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>Multi-Jointed Robot support</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>0</td>
</tr>
</tbody>
</table>

778
On design aspect, there are some advantages that this tool simulates the robot motion before actually developing it, has parallel mechanism (Fork-Join / Reverse Fork-Join) to design dynamic models, and supports heterogeneous platform / hardware. On implementation aspect, it automatically generates source codes (such as C, C++ and Java) which the developer needs.

5. Conclusion

We propose MDD based modeling tool to design heterogeneous multi-jointed Robots. We can also quickly develop heterogeneous robots with model transformation algorithm and profiles. We are still research on the extension of UML notations and modeling & simulation to completely model the heterogeneous robots, and also on the extension of profiles to apply many other targets.

ACKNOWLEDGEMENT

This research is financially supported by the Ministry of Education, Science Technology (MEST) and Korea Industrial Technology Foundation (KOTEF) through the Human Resource Training Project for Regional Innovation(2009).

REFERENCES
