J. Internet Comput. Serv.
ISSN 1598-0170 (Print) / ISSN 2287-1136 (Online)
http://www.jics.or.kr
Copyright ⓒ 2019 KSII

Y Block Diagram as a New Process Notation in a GPS Manufacturer☆

Jung-Gyu Lee[1]　Seung Ryul Jeong[2*]

ABSTRACT

Company A should maintain myriad conversion tools for the purpose of making a geometric compilation of navigation maps. Company A is already using complex compilation tools, which are tailored to geographical areas and various GPS models. However, due to frequent requirement and personnel changes, there is an endless challenge for perfect tool configuration and multiple map consolidation. To solve this problem, Company A launched a process automation project using Graphviz, which is an open source workflow graph visualization software. Before implementation, they had to document their current map compilation processes and then match it with the applicable conversion tool. For effective representation of process controls, a new graphical process notation is designed, i.e. Y Block diagram. The authors will compare Y Block diagram with other process notations and explain why Y Block diagram is more useful for tool based business processes such as digital map generation processes.

☞ keyword : Process Notation, Workflow Management, Digital Map, Graphviz, GPS, Y Block

1. Introduction

Company A, a GPS manufacturer, broadly, has two lines of development processes. One is for sequential digital map conversion and compilation processes. The other is for embedded discrete software development which uses the before mentioned digital maps for navigation application. Authors are focusing on map conversion and compilation processes. For map compilation, company A should maintain a large number of software conversion tools. They are using several hundreds of compilation tools tailored to world-wide geographical areas and various GPS models. The processing sequence of the map conversion should be synchronized batch job processes with applicable tools.

It is very critical to get error free intermediate and final outputs such as the characteristics of the chemical process industry. However, due to frequent requirement and personnel changes, there is an endless challenge for perfect tool configuration and multi-layered map consolidation. To make matters worse, those conversion tools had been archived in the engineer's personal desktop computers. As a result, there were too many tool variances to manage multiple versions of geographical maps and GPS devices. That was the reason why company A had launched a process automation project. Before project implementation, they had to document their multiple, step-by-step procedure for matching the right conversion tool with the map compilation batch processes. For effective representation of tool alignment with process control, a new graphical process notation is designed, i.e. 'Y Block notation (or diagram).'

Through our empirical research, we have found that the Y Block diagram model has been used successfully to document standard conversion processes, especially concentrated on tool configuration which is matched for geographical and GPS model specifications. Therefore it provides strong support to design the automated workflow management (WFM). Company A had selected Graphviz as the WFM platform. Graphviz is an open source workflow graph visualization tool provided by AT&T. The next section will explain some research background and literature review for comparing the Y Block diagram to major business process notation diagrams. The authors will compare the Y Block diagram with other process notations and explain why

1 KCA, Seoul, 06502, Korea.
2 Graduate School of Business IT, Kookmin University, Seoul, 02707, Korea.
* Corresponding author (srjeong@kookmin.ac.kr)
[Received 29 August 2018, Reviewed 30 October, 2018, Accepted 30 December, 2018]
☆ A preliminary version of this paper was presented at ICONI 2017 seminar and was selected as an outstanding paper.

http://dx.doi.org/10.7472/jksii.2019.20.1.125

the Y Block diagram is more useful, especially for tool based business processes such as digital map application. This is followed by benefits for applying the Y Block diagram to Graphviz.

2. Background

The digital map compilation processes of company A includes over 200 job steps to make a set of final binary map files. There are multiple sets of map files on thirteen map scales for each GPS devices. It means that the company uses several thousand map conversion tools. Usually, the government provides the original baseline map (1:5,000 scales) for the domestic market. And then, company A should manipulate the components of map elements including line, shape, color, road link and put various physical attributes like points of interest, buildings, cost values (i.e. speed limitation, average speed) on each link id and etc. Even when the whole map compilation process becomes extremely complex, it should still be error-free and completed in time to meet release deadlines. In order to avoid errors, engineers should align proper conversion tools for every step-by-step compilation process. However, there are several barriers to reaching this stage of error-free perfection:

- Due to frequent engineer replacement, specific project knowledge and experience is not always transferred between old and new personnel. Lack of vital knowledge and experience transferred between personnel due to frequent engineer replacement.
- Tool variance between standard conversion software and derivative versions archived in engineer's desktop PC.
- Insufficient time to maintain tools due to clients' delivery pressure.

Ironically, the above issues worsened when company A hired outside partners to speed up the compilation process. Also, ad hoc requirements from customers created several non-standard compilation tools. Tool variance became the main source for compilation errors and delayed cycle time because novice engineers were not able to articulate the best tools to apply.

The foregoing reason moved company A management to consider developing a new map compilation process. So, the process automation by WFM tool was implemented according to multiple geographical regions and GPS models. FYI. Each GPS model is using different user interface and map shape for differentiation marketing policy according to the selling price. But, they are sharing the same node and link data structure.

Prior to the introduction of the process automation, it was necessary to document the as-built process model in company A. The Y Block diagram was newly designed by R&D team because they believed legacy notations were not good enough for their tool based procedures. They redefined the requirement of the new schematic notation model with these features:

- To describe the hierarchical abstraction according to process levels.
- To mark the input and output filenames of each process.
- To describe the exact tool name and its version per each conversion activity. (Most important requirement for company A management)
- To identify the person responsible for carrying out each conversion process.

R&D team had selected Graphviz, an open source workflow graph process tool, as the process automation platform running on Apache Hadoop architecture. The newly designed Y Block diagram provided a valid benefit to implementing the automation of map compilation process ahead.

3. Literature Study

From a cognitive point of view, diagrammatic representation for the business process is easier to understand when compared to a textual or numerical only presentation [1]. Schematic models for implementing the business process on an information system are mainly being used by DFD (data flow diagram) and ERD (entity relationship diagram). DFD is proper to represent the flow of data through an information system. ERD focuses on the associations and dependencies between entities, not on business processes. Although both represent a model for expressing the data model of the entity, it is inappropriate for expressing the sequential steps of engineering or business processes.

Some other approaches related to the process notation are BPMN (business process management modeling notation), UML (unified model language) Activity Diagram (AD). BPMN is a de facto standard that is capable of representing nearly any conceivable business process. However, Fernandez insists that BPMN is still too complex for business domain people to represent business processes even if it is relatively easier than UML AD [2]. Likewise, UML AD is not easy to draw without understanding the Unified Model Language and it lacks the representational effectiveness needed for business processes.

Kock introduced the concept of communication flow orientation(CFO). He ranked 7 CFO levels from the lowest level communication flow model, i.e. standard flow chart to Functional Flowchart, UML AD, UML Use Case Diagram, UML Communication Diagram, Data Flow Diagram, and the highest level communication flow model, i.e. Communication Flow Diagram. He explained that a higher communication flow oriented model is more effective for IT system implementation rather than a functional flowchart. The key message is that an easier documentation model is also useful for understanding the concept of business models within the context of IT implementation [3].

According to Bibliowicz, correctness and comprehensibility are the two basic requirements for a system’s conceptual model[4]. He recommends OPD (Object Process Diagram), but OPD also has a higher level of semantic complexity for describing graphic symbols. Another approach, the Markov process, which is a transition system, uses a simple combination of symbols like states and arcs. Though it is effective in predicting the probability of process logical errors, it is not effective for aligning right resources (i.e. tool lists). This is why company A considered designing another process notation model. While the Y Block diagram lacks logical node and link concepts that are common for general business process notations like Petri net or BPMN, this chart provides a clear and efficient representation with tool configuration, especially for geometric element conversion or compilation on digital maps.

Workflow management systems use a large variety of process/visual languages and concepts based on different paradigms. Most available products use a proprietary visual and process language rather than a tool-independent language [5]. Aalst and Hofstede also mentioned that one of the reasons for the lack of consensus on the consolidated language was the variety of different specifications in which business processes are otherwise described. As a result, they proposed the following four different perspectives of workflow specification:

- Control-flow perspective
- Data perspective
- Resource perspective
- Operational perspective

Resource perspective is related to a human or a device resource, whereas the Y block diagram addresses software resources, especially compilation tool resources in digital map application. Table 1 shows the key characteristics of major graphical and/or process languages including the above-mentioned tools. Thirteen graphical and/or process languages are presented in table 1. Table 1 does not show all the graphical and/or process languages, but most of the popular market notations are covered. These process notations mainly focus on process activities. But company A required a specialized notation scheme to verify and validate a sequence of conversion tools rather than a sequence of activities.

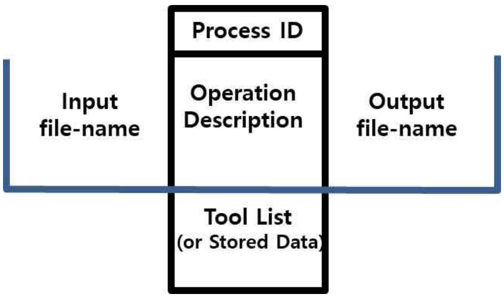
4. Y Block Diagram

4.1 Basic Notation

The Y Block diagram gets its’ name from its resemblance to hands above the shoulder posture of the letter Y. The diagram is composed with the Y Block and arc (arrow) only. The Y Block diagram has 5 parts as shown in figure 1. They are the following:

- Process ID
- Input filename
- Output filename
- Conversion job description (name) matched with corresponding process ID
- Conversion tool name or list (or stored data)

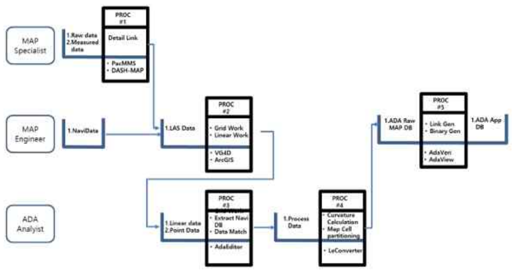
The conversion tool name with version information is the most important piece of data for aligning map compilation processes. From the reader's POV, the left arm identifies the input filename. Likewise, the right arm is used for the output filename. A drawing of the left or right arm corresponds to the physical location of the files. If the input and output files are located in the same physical storage, Y block can have dual arms. Each Y block diagram has only one process id. However, the other parts may have multiple entities, e.g. multiple input files, output files, conversion job names, and conversion tools. The arc connects the Y Blocks, directing from output to input. Multiple inputs and outputs are allowed. The arc shows the conversion processes only, and there is no specific drawing limitation for the arc. Either a line or curve is acceptable, but not both. The arc can be attached to either arms or the bottom of the body, but arm attachment is preferable.



(Figure 1) Y Block Diagram Legend

4.2 Process Notation

The Y Block diagram adapted its schematic idea from IBM's HIPO (hierarchy plus input process output) chart. However, the Y Block represents input/out filenames, and at the very least, process description and conversion tool names. Figure 2 demonstrates a pseudo-conversion process with multiple Y Block diagrams. As a role & activity diagram, the Y Block diagram can be aligned by process owners. Each process owner is responsible for the integrity of data files and conversion tools. There is no feedback cycle in the Y Block process. The Y Block continuously runs from left to right and does step-by-step conversions. If errors develop, tool errors should be fixed, and then the conversion process should be restarted from the previous process breaking point.



(Figure 2) Y Block Role-Activity Chart

4.3 Configuration Table

The Y Block methodology has several complementary documentation formats. For example (1) process description tables, (2) conversion tool tables, (3) input template, (4) output template are all compatible with Y Block. Table 2 shows the process description table and conversion tool table. The process description table explains the attributes of each process's IDs. Most of the attributes can be described fully in the Y block diagram, but the table may have additional information for standard work hour and quality level (e.g. 3.0 sigma). Conversion tool tables articulate the control number of conversion tools, tool name, execution filename, applicable process ID, process name, and initial build date, recent update date, current version, developer's name, and finally the runtime environment (or IT constraint).

Thanks to the Y Block diagram, company A can clearly define the configuration of conversion tools and the responsibility of each process jobs. And they easily replicate the new conversion process with the given conversion process and reuse those tools with a lower error rate. When the process conversion identifies a failure, engineers can locate the exact place of error break and analyze the geometry attributes relating to errors. Also with a configuration table like table 2, engineers can calculate the cycle time improvement and enhance other key performance indicators, including quality measures.

After finishing the above mentioned Y Block documentation, company A can easily apply map conversion toolsets to proper compilation procedures graphically enabled by Graphviz WFM.

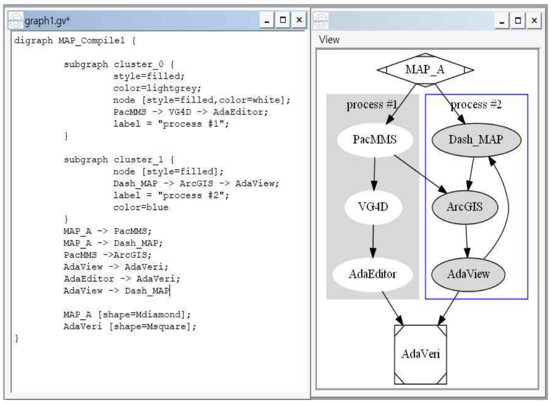
5. Graphviz Implementation

5.1 What is Graphviz?

Graphviz(graphviz.org) is an acronym for the graph visualization software, an open source graphical process tool by AT&T. The user can use 'dot' script language to draw nodes and edges (directional or non-directional). Graphviz provides various API and program libraries for C#, .Net, Java, Python, Ruby, R and etc. Company A selected Graphviz as the process control platform to get the following advantages:

- To detect errors by the color change of nodes (job steps) while running the batch job for map conversion.
- To restart batch jobs from the breaking point after the resolution of errors.
- To support parallel processing of Windows or Linux Hadoop environment.
- To get a best practice model to implement the multi-set for the map compilation processes.
- To copy the error-free conversion process scenario for another map version.

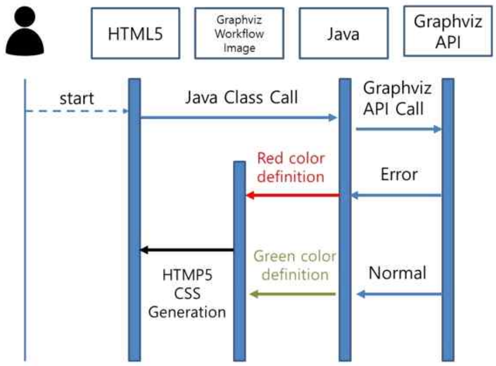
Figure 3 presents a sample of dot script language and its interactive graphical presentation. When the code declares "digraph" syntax, it means the edge between nodes is directional. Most of the 'dot' script codes are easy to understand. In the drawing, you can understand the two subgroups, the syntax for the shape of nodes, as well as, the color of nodes, edges, and labels. The edge direction is coded with "->" symbol.



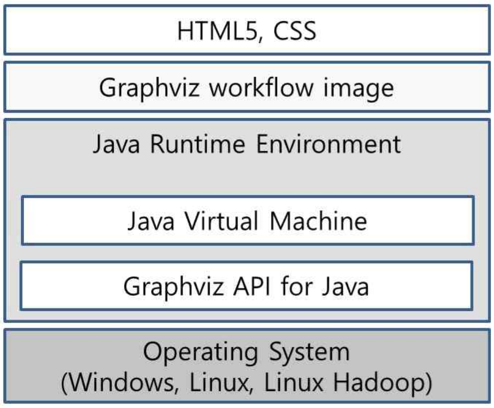
(Figure 3) tool mapping on Graphviz nodes(pseudo model)

5.2 Automated MAP Compilation Logic

Figures 4 and 5 are some diagrams for company A's system architecture for the automatic MAP conversion process. You may see the sequence diagram for the program running and the block diagram of software architecture. The basic user interface is running on the standard HTML5 web browser, Java logic, which classifies interlock with Graphviz API. Conversion tools are integrated with Graphviz API. In order to reduce computing time, conversion logic and data may run on Linux Hadoop platforms. Company A has multiple product lines of geometric maps. So, multiple batch job statuses are simultaneously displayed and managed on multiple HTML5 sessions.



(Figure 4) Graphviz UI Sequence Diagram

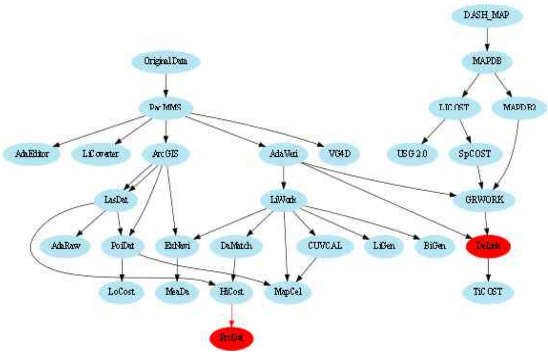


(Figure 5) Block Diagram for Software Architecture

Due to company A's security policies, the actual UI images of the Graphviz map conversion process are not available for publication. However, the conceptual representation of the process diagram is very similar to figure 6. Company A has dozens of map conversion processes because they are providing many geographic maps for many different clients in the global GPS market.

In the past, there was a possibility that matters of quality control would occur, as an individual engineer performed his/her part of the map conversion job by step-by-step procedure randomly with his/her tools. After the introduction of Graphviz workflow, standard conversion tools with the correct configuration were integrated with control processes. As a result, the possibility of conversion errors was dramatically reduced when the whole process was checked and set up in advance. Also, a new conversion process for a new GPS system can be easily referred and replicated from the given error-free conversion process model. Visual representation of error occurrence on Graphviz platform is very useful for an engineer to check the progress of conversion work. There is no longer a need for engineers to be put on standby in the middle of the night to monitor the conversion process because engineering can monitor the progress remotely through a web console.

Authors utilized a simple pseudo diagram for depicting company A's map conversion process because the process has a very similar diagram pattern to figure 6. But total nodes are about 200 in the real situation. In the drawing, the node color may change to red if an error develops; otherwise a green color indicates normal operation. When an error occurs, engineers can check the error log by mouse click and review the functional process of breaking process.



(Figure 6) Run-time workflow UI for automated map compilation (pseudo model)

6. Conclusion

The Y Block diagram in company A is a kind of mutant process diagram, but it is a relatively useful configuration tool for the map compilation application. In summary, the Y Block diagram is good for presenting the map conversion process in GPS Map company A. By using the Y Block notation, company A had smoothly implemented the automatic process control with Graphviz platform. Without the Y Block diagram, the engineer may find it difficult to match proper conversion tools per every conversion node on Graphviz. The Y Block diagram, another optimized notational model, is meaningful in the real world even if 'state diagram' or BPMN, which were mentioned in the literature study, is still dominant in process notation discipline. Some methods of notation are best suited for process description, while other methods of notation are the better fit for resource (i.e. tools) configuration, such as the Y Block diagram.

Whenever the limitations of standard notation models are encountered, people's creative endeavors to solve this short-coming by the development of new and relevant models should be respected as advancement. Of course, the Y Block diagram inherited a legacy from other process notation models, e.g. the HIPO diagram. The need for developing Y Block diagram came from Graphviz

(Table 1) Major Graphical and Process Language.

Process Language	Proposed by	Characteristics	Ref.
Simple BPMN (Business Process Modeling Notation)	Fernandez, 2004	• Consisted of about 10 simple symbols. • Task(activity), Event, Gateway, Swim-lane, Arc. • The most popular process language.	[2,6,7]
Petri Net	Carl Adam Petri, 1962	• Supported by ProM(process mining tool) • Transition, Place, Arc • The place must position between transitions. • Process control with token marking and firing operation.	[8]
YAWL	Workflow patterns initiative, 2003	• Enhancement of Petri net expressiveness. • Task, Condition, Arc. • Cancellation region • Syn. Support for multiple XOR/And-split/join.	[9]
EPC (Event-driven process chain)	August-Wilhelm Scheer, 199x	• ERP optimization: ARIS(Architecture of Integrated Information Systems) support • Function, Connector, Event, • Not allowed for OR/XOR-split of Events • Single Arc input and output to/from a function • No direct link b/w functions, arcs.	[10]
Casual Nets (C-net)	W.M.P. Aalst, 2011	• Designed for process mining. • Nodes for activities, Arcs for the causal relationship. • The binding of input and output arcs represents the split/join of activities.	[11]
Process Tree	-	• Process notation with process algebra. • Tree structure started from a root to branches(operators) and activities(leaves) • Designed for sound process model in process mining domain.	[12]
UML Activity Diagram	OMG, 1997	• Similar to the flowchart, easy to represent concurrency. • Bar symbols for split and join. • Note box for representation in detail. • Swim-lane is possible.	[13,14,15]
IDEF3 (Integrated DEFintion for Process Description Capture Method)	US Air Force, 197x	• From IDEF0 to IDEF1, to IEDF1x, and finally to IDEF14. • IDEF3 for business or system process modeling. • UoB(Unit of Behavior) for an atomic business process.	[16]
VSM (Value Stream Mapping)	Toyota, 198x	• Came from Toyota LEAN manufacturing methodology. • Initiated by Taiichi Ohno & Shigeo Shingo. • To verify and eliminate the non-value added processes. • High-level representation.	[17]
State diagram	Markov, 1906 Shannon, Weaver, 1949 OMG, 1997	• Initially designed for memory-less probability calculation by Andrey Markov in Russia. • Utilized for Finite State Machine. • UML statechart adopted the legacy of State diagram	[18,19]
Transition System	Similar to State Diagram	• The most fundamental process modeling notation. • Black line circle(state) and arc(transition) symbol. • Applicable for Finite-State Machine	[19]
BPEL4WS	OASIS, 2004	• Language specification of both executable and abstract business process through web service. • To define business processes using a XML-based language. • No standard graphic notations. It is not visual languages, but process languages.	[20]
XPDL	WfMC, 1998(V1.0)	• To interchange business process definitions between different workflow products. • Supported for both the graphics and the semantics. • The best file format for exchange of BPMN diagram • Contains elements to represent the graphical aspect of a process diagram such as the X and Y position of nodes)	[21]

(Table 2) Unit Process and Tool Specification

Process Table Example

Proc #	Proc. Name	Prior Proc.	Next Proc.	Input Data	Out Data	Proc. Owner	Standard Cost (hr)	Quality Index\ (Sigma)	Tool name
1	Detailed Route Works	start	Proc#2	Raw data Measured data	Las data	MSKIM	16	3.5	• PacMMS • DashMAP
2	Grid & Linear Works	Proc#1 Proc#a	Proc#3	Las data Navi data	Linear data Point data	HDKIM	32	3.0	• VG4D • ArcGIS
3	Extract Navi. Data & Data Matching	Proc#2	Proc#4	Linear data Point data	Process data	JDLEE	16	3.0	• AdaEditor

Tool Table Example

Tool ID	Tool Name	Filename	Proc#	Process Name	Initial build date	Latest build data	Latest Version	Owner	Run Time Environment
MT1	PacMMS	PacMMS-v19.exe	Proc#1	Detailed Route Works	10.4.1	15.2.1	V4.9	YIKANG	Max 1GB data
MT2	VG4D	VG4Design.exe	Proc#2-1	Grid Works	10.5.1	16.2.2	V3.0	JHKIM	BG batch
MT3	ArcGIS	ArcGIS.exe	Proc#2-2	Linear Works	11.6.2	17.12.1	V2.5	JHMIN	2D space only

implementation at that time. By using Y Block notation, company A can shorten the Graphviz implementation lead-time for enumerating the sequence of above 200 conversion tools per each map conversion process. If engineers specified the tool sequence according to resource matched with BPMN swim-lane, the process diagram might have a long spaghetti type drawing.

Graphviz-driven map compilation workflow increased the productivity for conversion process monitoring & control and shortened the modeling lead-time for new derivative map conversion processes. In the future, we will extend our research topics for the theory of tool-based process models including other resource attributes.

Reference

[1] Nergiz Ercil Cagilay and et al., "Performing and analyzing non-formal inspections of entity relationship diagram(ERD)," The journal of Systems and Software, Vol 86, pp.2184-2915, 2013. https://doi.org/10.1016/j.jss.2013.03.106

[2] H. Fernandez and et al., "SBPMN – An easier business process modeling notation for business users," Computer Standards & Interface, vol. 32, pp.18-28, 2010. https://doi.org/10.1016/j.csi.2009.04.006

[3] Ned Kock and et al., "Communication flow orientation in business process modeling and its effect on redesign success: Results from a field study," Decision Support Systems, vol. 46, pp. 562-575, 2009. https://dx.doi.org/10.1016/j.dss.2008.10.002

[4] Arich Bibliowicz and Dov Dori, "A graph grammar-based formal validation of object-process diagrams, " Software System Model, vol. 11, pp. 287-302, 2012. http://dx.doi.org/10.1007/s10270-011-0201-4

[5] W.M.P. van der Aalst & A.H.M. ter Hofstede, "YAWL: yet another workflow language," Information Systems, Vol 30, pp.245-275, 2004. https://dx.doi.org/10.1016/j.is.2004.02.002

[6] Michele Chinosi, Alberto Trombetta, "BPMN: An introduction to the standard," Computer Standards & Interface, vol. 34, pp. 18-28, 2012. https://dx.doi.org/10.1016/j.csi.2011.06.002

[7] Antonio Garcia-Dominguez and et al., "A Comparison of BPMN 2.0 with Other Notations for Manufacturing

Processes, ResearchGate, April 2012. https://dx.doi.org/10.1063/1.4707613.
[8] Seungchul Ha, Hyo-Won Suh, "A timed colored Petri nets modeling for dynamic workflow in product development process," Computers in Industry, vol. 59, pp.193-209, 2008. https://dx.doi.org/10.1016/j.compind.2007.06.016.
[9] http://yawlfoundation.org
[10] Anne Gross, Joerg Doerr, "EPC vs. UML Activity Diagram – Two Experiments Examining their Usefulness for Requirements Engineering, 17th IEEE International Requirements Engineering Conference. 2009. https://dx.doi.org/10.1109/RE.2009.30
[11] C-net: W.M.P. van der Aalst and et al, "Causal Nets: A Modeling Language Tailored Towards Process Discovery,"22nd International Conference, CONCUR, Sept. 2011. https://dx.doi.org/10.1007/978-3-642-23217-63
[12] J. Buijs, B. Dongen and et al., "A Genetic Algorithm for Discovering Process Trees," WCCI 2012 IEEE World Congress on Computational Intelligence, June 2012.
[13] Dan Pilone, UML Pocket Reference, 1st edition, O'Reilly, 2003.
[14] Object-Oriented Methods: A Foundation, UML edition(2nd edition), Prentice Hall PTR, Dec. 1997.
[15] Object-Oriented Methods: Programmatic Considerations, Prentice Hall, Jan. 1996.
[16] J. Ping and et al., "Research on business process simulation method of architecture based on IDEF3," IEEE(MSIE), 2011. https://dx.doi.org/10.1109/MSIE.2011.5707617
[17] 50Minutes.com, Value Stream Mapping: Reduce Waste And Maximise Efficiency, August 2017.
[18] D. Harel, "Statecharts: A Visual Formalism For Complex Systems," Science of Computer Programing, vol. 8, pp. 231-274, 1987. https://dx.doi.org/10.1016/0167-6423(87)90035-9
[19] J. Katoen and et al., "Three-valued abstraction for probabilistic systems," The Journal of Logic and Algebraic Programming, vol. 81, pp.356-389, 2012. https://dx.doi.org/10.1016/j.jlap.2012.03.007
[20] https://ibm.co/2wlEryU
[21] http://www.xpdl.org

◐ Authors ◑

JungGyu Lee
1986 B.S. in Mechanical Engineering, SungKyunKwan Univ., Seoul, Korea
1996 MBA in Sogang Univ., Seoul, Korea
2019 Ph.D. in Business IT, Kookmin Univ., Korea
2011~2013 Adj. Prof., Graduate School of KAIST Innovation & Tech. Management, Seoul, Korea
2017~Present: IT Auditor, KCA, Seoul, Korea
Research Interests: Business Process Innovation, Process Governance, Startup Business Process
E-mail: dominic@kookmin.ac.kr

Seung Ryul Jeong
1985 B.A. in Economics, Sogang Univ., Seoul, Korea
1989 M.S. in MIS, Univ. of Wisconsin, WI, U.S.A.
1995 Ph.D. in MIS, Univ. of South Carolina, SC, U.S.A.
1997~Present: Professor, Graduate School of Business IT, Kookmin Univ., Korea
Research Interests: System Implementation, Process Innovation, Project Management, Information Resource Management etc.
E-mail: srjeong@kookmin.ac.kr