Integration of queuing network and IDEF3 for business process analysis

Ki-Young Jeong
Engineering Management & Systems Engineering,
University of Houston Clear Lake, Houston, Texas, USA
Hyunbo Cho
Division of Mechanical and Industrial Engineering,
Pohang University of Science and Technology, Pohang, Republic of Korea, and
Don T. Phillips
Department of Industrial and Systems Engineering, Texas A&M University,
College Station, Texas, USA

Abstract

Purpose – The purpose of this paper is to provide a framework and prototype software to use IDEF3 descriptions as a knowledge base from which a queuing network (QN) analysis is performed to compute system performance measures as part of quick response manufacturing. This intends to help domain experts obtain informative quantitative performance measures such as resource utilization, waiting time, and cycle time without relying on a time consuming simulation approach.

Design/methodology/approach – A general open queuing network is used to extract the related resource information from the process knowledge captured by IDEF3 method. The relational database is used to integrate the open QN and IDEF3, which also improves the knowledge reusability. In addition, the performance of the open queuing network analyzer (QNA) is compared to the simulation through case studies.

Findings – The domain experts usually do not own much technical modeling knowledge. However, through this integration, it is found that they could obtain several meaningful system performance measures without simulation. They could also perform the diverse "what if" scenario analyses with this prototype without difficulties. It is another finding that the system performance measures generated by the open QNA are reasonably close to the values obtained from simulation, articularly when the system utilization is low.

Research limitations/implications – The open QN analysis used in this integration is not as generic as the simulation approach in terms of the modeling scope and capability. Hence, this integration supports only the exclusive OR (XOR) out of three junctions in IDEF3 grammars.

Practical implications – Some system analysis problems do not require a complex simulation modeling approach. Domain experts need a modeling tool to quickly obtain some system dynamics and insights. This integration framework suffices those requirements.

Originality/value – This paper describes the first attempt to generate informative system performance measures from the IDEF3 model using the open QN. It also offers practical help to the domain experts working in the system analysis area.

Keywords Simulation, Knowledge management, Business process re-engineering, Process management, Queuing theory, Modelling

Paper type Research paper

Business Process Management Journal
Vol. 14 No. 4, 2008
pp. 471-482

1463-7154
DOI 10.1108/14637150810888028

1. Introduction

IDEF3 is a descriptive process modeling method, which graphically represents the process knowledge of a given system in order to improve the communication between project members. However, since it does not provide any quantitative analysis for the system, some researchers tried to integrate the simulation with IDEF3 by generating a simulation model from an IDEF3 model to numerically explain the behavior of the systems. For example, KBSI (1995) and Resenburg and Zwemstra (1995) developed the mechanism to generate a WITNESS and a SIMAN simulation model from an IDEF3 model, respectively. Although these approaches have been widely used, the extensive computer running time for simulation has been considered as a main disadvantage of any simulation-based approach, which motivated this research. If an analytical method can be used as a substitute for simulation in some situations whose modeling objectives do not require simulation-level modeling efforts, why not we integrate that analytical method with IDEF3?

In this study, we selected the general open queuing network (GOQN) as a substitute for the simulation since it can measure the resource contention and its effect on the overall system through the resource utilization and the system cycle time, etc. In fact, the GOQN can be applicable to many real life cases such as a flow shop and a job shops, and business process improvement as addressed in Bitran and Morabito (1996) and Shanthikumar and Buzacott (1984). The GOQN is robust in terms of data requirement since it is defined by:

- mean and variance of entity's external interarrival time;
- mean and variance of entity's processing time at resource; and
- entity's routing between resources.

That is, it does not require any specific form of a distribution as in simulation. However, we recognize that there are some analytical limitations and difficulties in the GOQN approach. For example, if a finite buffer or a non-First-In First-Out queue discipline issue is critical in a given domain, it may not properly work as a substitute for simulation. However, as Law and Kelton (1991) pointed out, the model needs to capture the essence of the system for which it is intended without excessive details. Hence, if none of the finite buffer or the queue discipline is a critical constraint or if it can be appropriately abstracted to reduce the complexity without violating the modeling requirements and objectives, the GOQN can still work as a solution method. Hence, if a GOQN analysis is available within IDEF3 environment, analysts can have an opportunity to deploy it as a substitute for simulation, and obtain rapid results without a detailed and time-consuming simulation model.

2. Modeling views

It is important to recognize for further study that IDEF3 uses a process-centered modeling view (PCMV) to capture the process knowledge in a given system, and GOQN uses a resource-centered modeling view (RCMV) for a quantitative analysis. If we define a *process* as a set of sequenced time-related *activities* performed by *resource(s)* to provide a service to an *entity* flowing through a system, i.e. products or customers, the PCMV defines the sequence of *activities* from the *entity*'s perspective regardless of the *resource*, and then it provides the resource information to each *activity*. Hence, the same *resource* may be represented multiple times in a *process* with the same or different names. For example, a process plan in a manufacturing system is a good example of the PCMV where the process is defined first to determine a product manufacturing sequence, and then specific resource information is assigned to it. However, in the RCMV, each resource is first recognized and uniquely defined, then the input and output flows of the entities to and from the resource are defined. Hence, each resource can be represented only once in the RCMV. Many analytical modeling methods such as the Petri net and the GOQN use the RCMV.

Figure 1 shows the difference between two modeling views. The flow of the two *entities* – e1 and e2 – is represented as a solid arrow. The first diagram (a) shows a type of the PCMV representation where a sequence of the named circles represents a process defined for each entity. Note that, the name of each resource providing a service to each entity in an activity is denoted under the activity. The same resource R1 is repeatedly used in the different activities at each process. The second diagram (b) describes the same system from the RCMV perspective where resource name is uniquely represented in a circle, and all flows are represented between R1 and R2. In general, the PCMV is more intuitive and commonly used in a descriptive model while the RCMV is more popular in a quantitative model since it is easy to quantify the amount of flow between resources. Hence, to integrate the IDEF3 with the GOQN, we need to convert the PCMV-based knowledge in IDEF3 to the RCMV-based knowledge in GOQN.

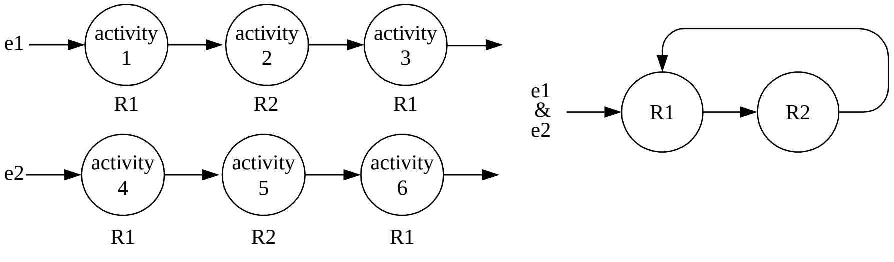
3. Overview and concept of a proposed framework

The conceptual architecture of the integration is shown in Figure 2, which shows the four modules; knowledge acquisition and representation (KAR), modeling view converter (MVC), database, and queueing network analyzer (QNA). The KAR module is composed of the IDEF3 and the queuing network (QN) graphics module.

A typical operation among the modules has the following sequence:

- the IDEF3 graphics module captures the process knowledge using the PCMV;
- the MVC translates this PCMV-based knowledge into the RCMV-based knowledge;
- the QN graphics module refines the RCMV-based knowledge; and
- the QNA performs the GOQN-based QN analysis in order to numerically explain the IDEF3 process model.

Keeping all knowledge in the database facilitates the knowledge reusability since all captured and processed knowledge is separated from the IDEF3 and the QNA, which can be easily accessed via Structured Query Statement. Hence, if analysts want to deploy another analytical tool, i.e. simulation later, this knowledge can be reused easily.



(a) Representation from PCMV **(b)** Representation from RCMV

Figure 1. Examples of two different modeling views

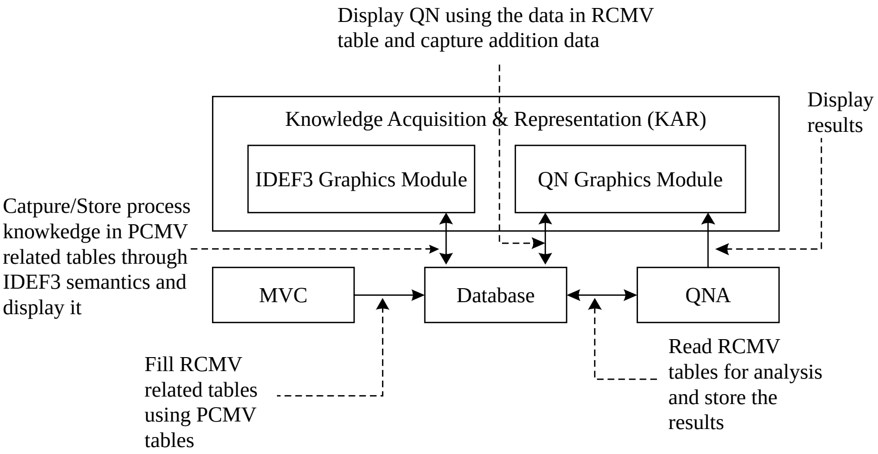


Figure 2.
Conceptual architecture of a proposed framework

Now, we are ready to explain each module. All italicized words refer to the objects or tables in the database module.

3.1 IDEF3 graphics module

Since IDEF3 plays an important role in our approach as a main information capturing method, it is useful to briefly review the concepts of the IDEF3. The IDEF3 is one of the integrated definition methods developed by the Information Integration for Concurrent Engineering program sponsored by the USA Air Force's Armstrong Laboratory (Mayer *et al.*, 1995). The primary goal of IDEF3 is to present a structured method by which a domain expert can capture the processes of a particular system. The process schematics of IDEF3 have been widely accepted as a medium for process description in industry (Mo and Menzel, 1998). The process schematics consist of the three main components:

1. unit of behavior (UOB);
2. junction; and
3. link.

A UOB captures information on what is going on in a system to represent a process or an activity. It is depicted by a rectangle with a unique label. Junctions in IDEF3 provide a mechanism specifying a logical branching of UOBs and introduce the timing and sequencing of multiple processes. Junction types include a conjunctive AND junction denoted by "&" and two disjunctive junctions: an inclusive OR denoted by "O" and an exclusive OR denoted by "X". However, in this paper, it is assumed that the process model does not include any inclusive OR and conjunctive AND due to its analytical complexity, and we believe that those junctions can be appropriately handled in the simulation environment. A link connects UOBs or "Junctions". Hence, any IDEF3 process schematics can be represented by $\mathbf{G}_{\text{IDEF3}} = (\mathbf{U}, \mathbf{J}, \mathbf{L})$ where $\mathbf{U}$, $\mathbf{J}$ and $\mathbf{L}$ are the set of UOB, junctions, and links, respectively. Table I summarizes the process schematics of IDEF3 within this paper's boundary.

Table I. IDEF3 process schematics

Name	Description	Symbol
Unit of behavior (UOB)	Capture information on what is going on in the system, which represents a process or an activity	name ID
Link	Represent temporal, logical, causal, natural or relational constructs between UOBs	→
Junctions	Specify a logical branching of UOBs	
	Fan-Out XOR: exactly one of the following paths will be activated	X
	Fan-In XOR: exactly one of the preceding paths have completed at a time	X

As a major user-interface, the IDEF3 graphics module captures a sequence of activities from the entity perspective through IDEF3 schematics, and this captured knowledge is stored at the appropriate IDEF3 related tables in the database. If analysts want the quantitative analysis for the process captured, they have to capture the resource information for each *activity* associated with UOB – to facilitate the data manipulation in the QN graphics module. Through this information processing, the IDEF3 related tables or called PCMV tables in Figure 3 such as *UOB*, *Process*, *Activity* and *Junction* are populated, and the QN related tables or called RCMV tables such as *Routing* and *Operation* are structured. Note that, we classify that *Equipment*, *Operator, and Product* as the common data objects or tables related to both modeling view.

3.2 Modeling view converter and database module

The role of the MVC is to populate the QN related tables using the data in the IDEF3 tables to facilitate the performance of the QN graphics module and the QNA. That is, the PCMV-based knowledge stored at IDEF3 related tables is transformed into RCMV-based knowledge in the QN related tables. For example, the *Activity* information is directly reused to construct the *Operation* due to the similarity between two objects. The *Routing* is constructed based on *Operation*, *Process* and *Activity* to

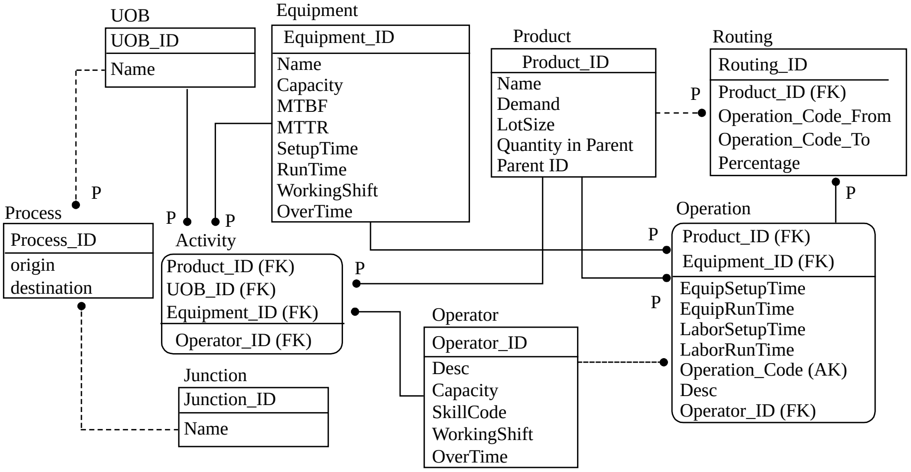


Figure 3. IDEF1X data model for facilitating a proposed framework

provide the probabilistic sequence of each operation for each product. The *Process* contains the sequence of *UOB* describing the interactions between product, resource and operator within *Activity*. In fact, the MVC is a macro-module working on the database to make the QN analysis-friendly data format.

The database design should be robust enough to support both PCMV and RCMV. One way to consider the robustness in a system design is to study its ontology, and reflect it in the relational database design since ontology provides the definition of the terminologies, objects, and relationship between them in the system. The IDEF1X data modeling method can be used for this purpose since it captures objects with attributes and the relationship among the objects in a given system. Figure 3 shows a partial IDEF1X data model for this study. The *Equipment*, *Product*, *Operator*, *Routing*, *UOB*, *Process* and *Junction* are defined as an independent object (rectangle) with attributes while the *Activity* and *Operation* are represented as a dependent object (rounded rectangle). An independent object can be identified itself while a dependent object is identified by the foreign key(s) migrated from the independent objects. The dotted line represents a non-identifying connection relation, meaning that the object can be uniquely identified without knowing the associated objects while the solid line represents the identifying connection relation, implying that an object is identified with its association. The dot without "P" represents zero, one or more relationship between objects while the dot with "P" denotes the one or more relationship.

All IDEF3 related information from $\mathbf{G}_{\mathrm{IDEF3}} = (\mathbf{U}, \mathbf{J}, \mathbf{L})$ is stored in the IDEF3 related tables. The *Activity* describes the interaction between resource and product within UOB, and the *Process* shows the sequence information conveyed by "Link" in IDEF3. Once this information is captured with data objects such as *Equipment*, *Operator* and *Product*, the QN information is computed by the MVC. The *Operation* provides the detailed specification for each operation performed at each resource, containing all operation information that describes "who (operator) handles what product(s) with what machines for what time." The *Routing* shows the probabilistic sequence of flow of products among resources.

The *Product*, *Equipment* and *Routing* provide:

- external inter-arrival time (demand data);
- processing time (service data); and
- routing information between resources (routing data), respectively, which defines a QN.

Note that, although the variance information regarding the inter-arrival time and the processing time are not shown in Figure 3, those values are captured in the QN graphics module.

3.3 Open queuing network analyzer

A QN can be described as $\mathbf{G}_{\mathrm{q}} = (\mathbf{N}, \mathbf{A})$ where $\mathbf{N}$ is a set of nodes representing resource and $\mathbf{A}$ is a set of arcs representing the direction of flow among nodes. The RCMV tables generated by the MVC provide all information for the frame of a QN. For example, the *Equipment* and *Routing* store a set of nodes and arc information, respectively. The major role of the QN graphics module is to complete both RCMV tables and common data tables such as *Equipment*, *Product* and *Operator* for the QNA. It captures:

- mean and variance of external inter-arrival times for each product (demand data) in *Product*; and
- processing time (service data) for each resource, *Equipment* and *Operator*; and
- the routing probability between nodes (resources) for each product in *Routing*.

The routing probability is stored at "Percentage" field in the *Routing* table.

3.4 Queuing network model and analysis

The GOQN, $\mathbf{G}_\mathrm{q} = (\mathbf{N}, \mathbf{A})$ can be solved by:

- mean and variance of entity's external inter-arrival time;
- mean and variance of entity's processing time at resource; and
- entity's routing between resources, and this information is stored at the *Equipment, Product*, *Operator*, *Operation*, and *Routing* tables.

Hence, the QNA solves the problem using this information based on the GOQN theory. For example, the inter-arrival time is estimated from "Demand" and "LotSize" in *Product*, "SetupTime," "RunTime" and other time information in *Equipment* or *Operation* can be used for the processing time. The "Percentage" in *Routing* represents the probabilistic routing probability.

Although, it may not easy to observe the direct information mapping from an IDEF3 to a QN, the information captured by IDEF3 was used to construct the QN frame. For example, the sequence of products among resources represented in the sequence of UOBs in PCMV tables provides the structure of routing in the QN, $\mathbf{G}_\mathrm{q}$. It also helped to develop the structure of the common data objects by capturing the lists of resource and products. Based on this frame, the QNA can solve the problem with mathematical formulae after the QN graphics module finalized the QN.

We briefly showed some formulae used in this module. Each node in $\mathbf{G}_\mathrm{q}$ is considered as a *GI*/*G*/*c* queue where the notation *GI*, *G* and *c* refers to general inter-arrival time, general service time and number of resource, respectively. Since, it does not require any specific distribution for inter-arrival and service times, the data collection efforts can be reduced compared to those in simulation. QN analysis consists of two steps:

1. decomposition; and
2. aggregation.

The decomposition computes the node-level performance measures such as resource utilization and sojourn time, etc. and the aggregation computes the system-level performance measures, such as system cycle time and total WIP in the system. Hence, information at *Product*, *Equipment* and *Operator* is used for all formulae in this section. The utilization for each resource j, $u(j)$, is represented in equation (1):

$$u(j) = \frac{\sum_k \lambda^k(j)}{\mathrm{TA}(j)} \tag{1}$$

where $\lambda^k(j)$ is all workload caused by product k at resource j, computed using the routing, demand and processing time information stored in *Routing*, *Product*, and *Equipment*, *Operation*, respectively. $\mathrm{TA}(j)$ is total available time at resource j. If $u(j) \geq 1$, the system is infeasible, saying that the steady state analysis is not possible.

The prompt computation of equation (1) is another advantage of QN over simulation. For example, it is not easy to detect the model's feasibility in a large-scale simulation model development due to the data and modeling complexity. Hence, the model is likely to be developed without full consideration of feasibility, which often results in considerable WIP accumulation in a queue. The major performance measures in GOQN are approximated using the results in *M/M/c* queue for which all-exact solution forms are known. For example, the expected waiting time for any product at resource *j*, $E(Wq_jGI/G/c)$, is approximated using the waiting time at *M/M/c* queue as in equation (2):

$$E(Wq_jGI/G/c) \approx \left(\frac{Ca_j + Cs_j}{2}\right)E(Wq_jM/M/c) \qquad (2)$$

where Ca_j and Cs_j are squared coefficient of variation for interarrival time and service time, respectively. Again, the *Product* and *Equipment* provide values for this computation. Other node-level performance measures can be computed using the Little's (1961) formula once $E(Wq_jGI/G/c)$ is computed. The key aggregation procedure is to compute the system cycle time given by:

$$E(CT^k) = \sum_j E\left(N_j^k\right)(E(Wq_jGI/G/c) + s^k(j)) \qquad (3)$$

where $E(N_j^k)$ and $s^k(j)$ represent the expected number of visits to *j* and process time at *j* for product *k*. If the lot size is considered, the processing time at resource *j*, can be represented as:

$$s^k(j) = st(j) + Q^k t^k(j) \qquad (4)$$

where $st(j)$, Q^k, $t^k(j)$ denote lot size independent setup time, product *k*'s lot size and processing time of individual piece in the lot at resource *j*. That is, by using equations (3) and (4), the impact of setup time and lot size on the cycle time can be estimated. Readers are encouraged to refer to Bitran and Morabito (1996) for a detailed computation procedure regarding all terms in equation (1) and (3).

When the $\mathbf{G}_{\mathrm{IDEF3}}$ has the same structure as the $\mathbf{G}_{\mathrm{q}}$, i.e. a resource is used only once across the $\mathbf{G}_{\mathrm{IDEF3}}$, each UOB is considered as a single node in $\mathbf{G}_{\mathrm{q}}$, and the sojourn time computed by equation (3) is the same as the activity cycle time in UOB, there is no need for modeling view converting.

4. Prototype software development

Software "SmartQueue" was developed to implement the proposed concept and framework. It provides a user-friendly graphic interface for the IDEF3 process descriptions and the QN analysis.

The QN diagram is built from an IDEF3 schematic diagram, and all IDEF3 syntax is reused for the QN diagram. For example, an XOR junction and UOB rectangle are reused to represent the probabilistic routing and a resource node, respectively. All artifacts constructed in "SmartQueue" are stored in a pool for reuse. It also allows users to build their own sub- QN (template) in a library. Users can retrieve this template when they want to expand or make the QN model in more detail. Once this template is used, the users are supposed to connect the template with the existing QN.

Figure 4 shows a screen shot of a template with its entity queue information window that is a part of the QN graphics module. It is noted that, a resource name is represented inside each rectangle. The "Entity Queue Information" window shows all the required input data for each product at each node in a template, which includes:

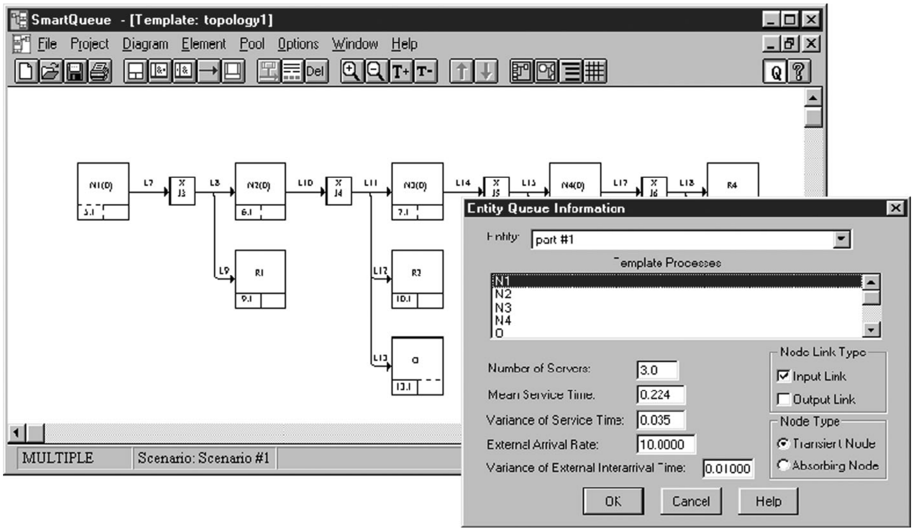
- number of servers;
- mean service time;
- variance of service time;
- external arrival rate; and
- variance of external interarrival time.

In addition to this input data, the node type information should also be specified for QN analysis. For example, any arriving product enters the template only through input node(s) and leaves it through output node(s). The input and all other nodes except output node(s) are considered as a transient node, and any output node is considered as an absorbing node. The template is connected to the existing QN through input and output nodes. Once all information is provided, the "SmartQueue" performs QN analysis based on the GOQN theory.

5. Case studies

A machining shop was used as a case study to implement the proposed concept with "SmartQueue" software. Once we obtained the results with "SmartQueue," we developed the simulation model from the same database used in the "SmartQueue" to compare the performance of the QNA with the performance measures in the simulation. The system cycle time – time for each product to spend in a system – is used as a major criterion.

The machining shop produces gearboxes used for automobiles. This shop are handling about 50 different part types of gearboxes manufactured through various operations including metal cleaning, cutting, lining, drilling, grinding, welding, pressing, heat and



Entity: part #1

Number of Servers: 3.0

Mean Service Time: 0.224

Variance of Service Time: 0.035

External Arrival Rate: 10.0000

Variance of External Interarrival Time: 0.01000

☑ Input Link

☐ Output Link

● Transient Node

○ Absorbing Node

Figure 4.
Template and node information in "SmartQueue"

chemical treatment, and inspection. Among these, this shop performs only metal cutting, drilling, grinding, milling, and pressing related operations while vendors and other shops perform remaining and other operations such as cleaning, welding, heat, and chemical treatment, etc. The shop manager wants to reduce the system cycle time via the setup time reduction and re-layout of resources such as equipment and operators. New equipment may be purchased if required for setup time reduction. However, before making any decision, they decided to analyze current shop performance as a first step.

Since, the IDEF3 syntax in "SmartQueue" can provide a visualized process model, it was used to improve communications between shop managers and the project team. Through the IDEF3 model, we captured 35 different equipment types performing all the operations for 50 part types across all the facilities, each of which will eventually correspond to an individual node in the GOQN theory. Inside the machining shop, the 11 equipment types out of 35 were used to provide diverse operations. It was also observed that the routing between this equipment was not continuous. For example, the products may leave and revisit the machining shop in the middle of the whole manufacturing process since most of cleaning and chemical-related operations are performed at other shops and vendors. Eventually, all the finished products go through the non destructive test (NDT) operation, and are delivered to the final assembly shop located at other areas if the NDT result is successful. Otherwise, products need additional steps or they are destroyed. Since, we focused on the machining shop analysis, all other operations beyond this shop were considered as a time-holding block with an infinite capacity.

Table II shows the major equipment information in the shop; the average setup time per lot and the run time per individual piece in a lot. It should be noted that some information was masked to protect the company proprietary. This shop operates two 8-hour-shifts per day in which 1 hour is used for lunch or break. It also deploys five different operator teams responsible for operating the equipment as seen in the last column. The Deburr and press team have two members while all others have one member per shift. Figure 5 shows an annual demand distribution for all parts. The minimum, average and maximum values are 20, 450.52 and 5,406 units, respectively. The lot size for each part varies from 3 to 98 with average 14 units. To accommodate the variations existing in the demand and process time, the squared coefficient of variation for interarrival time and process time is assumed to be 0.3.

Once all the data were collected, the QNA in "SmartQueue" was executed, and the corresponding simulation model was created using the Enterprise-Dynamics

Equipment	No. of equipment	Setup time/lot (h)	Run time/piece (h)	Operators
Auto drill	1	15.10	8.20	Auto drill
Press	1	23.75	4.75	Press
Booth	2	3.38	10.07	Deburr
Drill press	1	4.33	11.67	Press
Lather	1	22.37	3.77	Press
NDT	1	14.48	3.07	Inspector
Semiauto drill	1	42.83	5.39	Press
Laser cutter	1	12.80	5.32	Laser
Manual shear	1	18.33	3.33	Manual shear
Milling	1	43.00	7.00	Press

Table II.
Equipment information

simulation library (Enterprise Dynamics, 2005) from the relational database. It was observed that a single run of simulation took about 15 minutes for two-year length. Figure 6 shows the comparison result of system cycle time for each part type in which the cycle time from simulation is the average of five runs to filter variations.

The average cycle time discrepancy between two methods was 6.22 percent using the following formula:

$$\text{Discrepancy (percent)} = \frac{|\text{SmartQueue} - \text{Simulation}|}{\text{Simulation}} \times 100 \qquad (5)$$

The shop managers were convinced with the fact that they could directly access the relational database to develop a simulation model if the future modeling objectives require more detailed analysis than done in the QNA. In practice, this knowledge reusability was considered to provide flexibility in performing an analytical modeling project. Additional tests were performed to show the effect of resource utilization on the performance of the QNA. Low utilization- and high-utilization cases were created based on this case study data. The same comparisons were performed for each case, and the results were summarized in Table III. It is noted that, the original case study is denoted as the medium

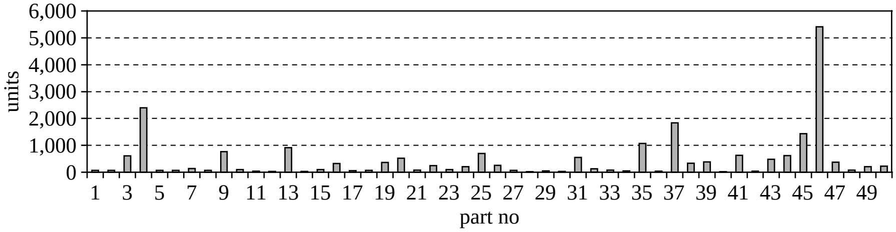


Figure 5.
Demand information

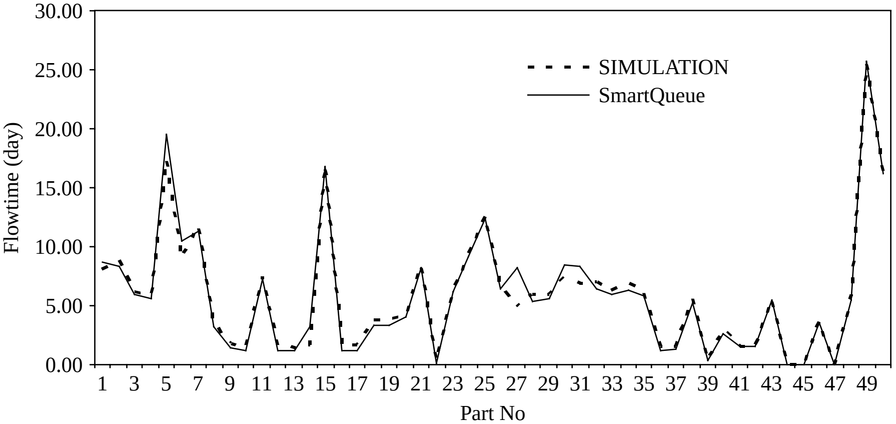


Figure 6.
Cycle time comparison between simulation and SmartQueue

Table III.
Result comparison

Case	Utilization (percent)	Simulation	SmartQueue	Discrepancy (percent)
Low	28	5.91	5.84	1.18
Medium	45	7.56	7.09	6.22
High	68	8.21	7.45	9.26

utilization case. According to Table III, the QNA underestimated the system cycle time, and the discrepancy between two methods increased as the resource utilization increased. These results are consistent with Desruelle and Steudel (1996) and other researches. Therefore, users need to consider the requirements and objectives of the modeling project before deciding an approach. Alternatively, users can use the QNA first, and then build the simulation model if additional analytical requirements are added.

6. Conclusions and further studies

The concept of transforming IDEF3 model into a QN model was provided and implemented through "SmartQueue" in order to show the feasibility to improve knowledge reusability and to add quantitative analysis capability to the domain knowledge descriptions captured in IDEF3. Within the scope of the case study, the accuracy of the QNA compared to simulation was reasonable in case of the moderate resource utilization. The integration and knowledge reusability through an independent relational database was considered to improve flexibility in choosing an appropriate analytical approach since we can avoid the times and efforts in developing and executing simulation models if the corresponding QN can satisfy the objectives of the modeling work. More research may be required to improve the capability of the QNA, i.e. computing an optimal lot size for each part to minimize cycle time. Another possible extension of this research is to develop a hybrid approach where both QN and simulation's advantages are integrated.

References

Bitran, G. and Morabito, R. (1996), "Open QN: optimization and performance evaluation models for discrete manufacturing system", *Production and Operations Management*, Vol. 5 No. 2, pp. 163-93.

Desruelle, P. and Steudel, H.J. (1996), "A queuing network model of a single-operator manufacturing work cell with machine/operator interference", *Management Science*, Vol. 42 No. 4, pp. 576-90.

Enterprise Dynamics (2005), *Reference Guide 4Dscript*, Enterprise Dynamics, Maarssen.

KBSI (1995), *ProSim™ Automatic Process Modelling for Windows, User's Manual and Reference Guide Ver. 2.1*, Knowledge Based Systems Inc., College Station, TX.

Law, A.M. and Kelton, W.D. (1991), *Simulation Modeling and Analysis*, McGraw-Hill, New York, NY.

Little, J. (1961), "A proof for the queuing formula: $L = \lambda W$", *Operations Research*, Vol. 9, pp. 383-9.

Mayer, R., Mensal, C.P., Painter, M., Dewitte, S., Blinn, T. and Perakath, B. (1995), "Information integration for concurrent engineering (IICE) IDEF3 process description capture method report", Interim Technical Report, Knowledge Based Systems Inc., College Station, TX.

Mo, J.O.T. and Menzel, C.P. (1998), "An integrated process model driven knowledge based system for remote customer support", *Computers in Industry*, Vol. 37, pp. 171-83.

Resenburg, A.V. and Zwemstra, N. (1995), "Implementing IDEF techniques as simulation modelling specifications", *Computers & Industrial Engineering*, Vol. 29, pp. 467-71.

Shanthikumar, J.G. and Buzacott, J.A. (1984), "The time spent in a dynamic job shop", *European Journal of Operational Research*, Vol. 17, pp. 215-26.

Corresponding author
Ki-Young Jeong can be contacted at: jeongk@uhcl.edu