Enterprise Asset Management as a Flow Machine

Sabah Al-Fedaghi and Nourah Al-Huwais

Abstract—**Enterprise Asset Management (EAM) is a broad term for software that provides a way to view company-owned assets holistically, where the goal is to control and proactively optimize operations for quality and efficiency. According to some published literature, knowledge is currently lacking regarding how to model EAM processes so they can be made ready for computerized deployment. This paper applies a new modeling technique built on systems of things that flow, to model EAM processes systematically. This flow-based modeling method is applied to a case study in a real enterprise that uses IBM Maximo. The resulting model points in a promising direction for EAM.**

Index Terms—**Conceptual model, Enterprise Asset Management, IBM Maximo, process.**

All *Asset management* is vital to the performance of most enterprises and is regarded as an essential process in many organizations. While asset management itself is an old discipline, as we enter a new century, it is attracting a great deal of interest because of advancements in technology that force a constant evolution of practices, among other factors. "The practices of asset management are constantly changing. New practices constantly emerge, either replacing the old or making current practices more complex" [1]. It is reported that companies practicing asset management lowered their annual costs by nearly 20% on average [2].

Asset Management deals with the acquisition, use, maintenance, modification, and disposal of critical assets and properties [1], including information, finances, competence and other intangibles insofar as they relate to asset management decisions [3]. ISO 55000 defines Asset management as the "coordinated activity of an organization to realize value from assets." According to Lingamaneni [4], an EAM *process* is "a set of linked activities and the sequence of these activities that are necessary for collectively realizing asset management goals."

In the context of this paper an *asset* is something of value to an organization. "Assets are often conceived of as a 'stock' from which a 'flow' of benefit is derived" [5]. Assets may be grouped into clusters of categories such as tangible/ intangible, physical/natural, current/deferred/fixed and performing/nonperforming [5]. Assets have attributes such as productivity, utility, security, susceptibility to risks, maintainability and control [6].

Enterprise asset management (EAM) is a broad term for software that provides a way to view company-owned assets holistically, where the goal is to control and proactively optimize operations for quality and efficiency [7]. The objective is to minimize the whole-life cost of assets and other critical factors such as risk or business continuity. EAM deals with the optimal management of asset systems and their life cycles to achieve value-for-money in the selection, design, acquisition, operation, maintenance, renewal, and disposal of (especially, physical) infrastructure and equipment. EAM systems include maintenance management capabilities for an organization's physical assets, and provide features to track, manage, and analyze asset performance and costs through the whole asset life cycle. Because they are designed for an entire enterprise, they include functions like maintenance, inventory, procurement, engineering and project management, accounting, operations, safety and compliance, and support for strategic planning.

ISO standards 55000/1/2 represent an international consensus on what asset management is and what it can do to increase value generated by all organizations [8].

Experience of several organizations has shown that significant improvements in asset/service performance, reduced risks and cost savings can be achieved by applying asset management systematically across the enterprise [9].

A. EAM Modeling

An Asset Information Model is a model that handles the data that support asset management and provides all the data and information related to, or required for the operation of an asset [10]. Koronios et al. [11] group asset management modules into three main areas:

Business Management: strategic planning, risk, financial management, budgeting, ownership.

Engineering Management: assets usage life cycle, performance measures, planning, monitoring.

(The focus of this paper) Information Management (IM): information systems, data management, resources. This area is the support layer of integration between the business and the engineering domains, and its role increases over time with the evolution of information technology disciplines.

Building Information Modeling is the management of information through the whole life cycle of an asset, including underpinning the creation, collation, and exchange of shared models and data. "This helps to avoid information loss when progressing between different life cycle stages and transitioning management responsibilities" [9].

B. Problem

Knowledge is currently lacking regarding *how to model* AM [Asset Management] processes in a methodical and appropriate manner so the processes are streamlined and

Manuscript received April 30, 2018; revised June 12, 2018.

The authors are with Computer Engineering Department, Kuwait University, P.O. Box 5969, Safat, 13060 Kuwait (e-mail: sabah.alfedaghi@ku.edu.kw,cpe_nora91@outlook.com).

DOI: 10.7763/IJMO.2018.V8.667

optimized and ready for computerized deployment [4].

Accordingly, this paper aims to develop a new technique to improve modeling of asset management processes. A conceptual diagrammatic language is proposed based on the notion of *things that flow* (to be defined later). The model is called the Flowthing Machine (FM) model, and a further general objective of this research is to explore the application of FM to modeling new areas. FM is an extension of the input-process-output (IPO) model that has been used extensively in interdisciplinary applications and is described as one of the most fundamental and important of all descriptive tools.

According to Black & Company [12], "Model your business process graphically. Business process diagrams (or models) are excellent at showing gaps in the process or errors in your understanding". Additionally, in computer science, "it is almost impossible to model without a conceptual diagram to visualize the modeler's concepts and the system" [12]. Conceptualization here refers to an *abstract* (i.e., independent of whatever technology is used) representation of the things that exist in a certain area.

In this modeling context, a *process* is typically defined as the transformation of inputs into outputs (e.g., ISO 9000:2005). In FM, a process is modeled as an *abstract machine* that receives, creates, releases, transfers, and processes (in the strict sense of *changes*) *things*. Recent years have seen an increase in interest in modeling languages used for depicting *business processes*, e.g., Business Process Modeling Notation (BPMN) and Unified Modeling Language (UML). Such languages are tools for constructing conceptual models representing the static and dynamic aspects of a system and reflecting a certain portion of reality.

Little research has explored a systematic approach to modeling of EAM processes. Koronios et al. [11] suggest UML specification for handling issues specific to the Information Management domain. "Even though UML can be at the foundation of other standards used in other domains, the UML in itself is part of the IM domain, as it deals with issues specific to the IM area" [11]. Lingamaneni [4] studied modeling of asset management processes through workflows. There are many studies that have applied UML to EAM systems (e.g., an auto repair shop [13]), and the process of designing and modeling a maintenance management system.

For the sake of self-contained paper, the FM model is briefly reviewed in the next section. This model has been utilized as a tool in several fields, including software engineering, business processes, and engineering design [14-20].

II. Flowthing Machine Model

Diagrammatic modeling is needed at the level between "natural communication" (i.e., spoken language) and semiformal specification to provide systematic thinking about events in a context of operations. This paper uses a diagrammatic language that depicts *assets* as *machines* comprising five basic "operations": creating, releasing, transferring, receiving, and processing of *things*. Such a language can play a central role in facilitating understanding among all participants and as a first step toward developing policies and implementation plans.

The abstract machine is a diagrammatic schema that uses *flowthings* to represent all types of physical and nonphysical assets. Flowthings *flow* in the machine among basic stages in which a flowthing can be created, released, transferred, processed, or received (see Fig. 1). Hereafter, flowthings may be referred to as *things* and an abstract security process as a *machine*.

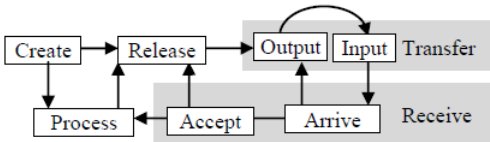

Fig. 1. Flow Machine.

The machine is the conceptual structure used to change or transmit things as they pass through stages, from their inception or arrival to their de-creation or transmission. Machines form the organizational structure (blueprint) of any security system. These machines can be embedded in a network of assemblies called *spheres* (e.g., airport, terminal, SSCP) in which the machines operate. The stages in Fig. 1 can be described as follows:

Arrive: A thing reaches a new machine.

Accepted: A thing is permitted to enter, or not (e.g., wrong credentials). If arriving things are always accepted, *Arrive* and *Accept* can be combined as a **Received** stage.

Processed (changed): A thing goes through some kind of transformation that changes it without creating a new thing (e.g., a passport is stamped at a checkpoint).

Released: A thing is marked as ready to be transferred outside the machine (e.g., a passenger is cleared to enter the boarding area).

Transferred: A thing is transported somewhere from/to outside the machine (e.g., from one airport to another).

Created: A new thing appears in a machine (e.g., a new passenger books a ticket).

The machine shown in Fig. 1 is a generalization of the typical *input-process-output* model used in many scientific fields. The stages in this machine are mutually exclusive. An additional stage of *Storage* can also be added to any machine to represent the storage of things; however, storage is not an exclusive stage because there can be *stored processed* flowthings, *stored created* flowthings, etc.

The notion of *spheres and sub-spheres* refers to network environments (e.g., the SSCP machine is within the *sphere of the arrival terminal*).

Multiple machines can exist in a sphere if needed. The machine is a subsphere that embodies the flow; it itself has no sub-spheres. *Triggering* is the activation of a flow, denoted by a dashed arrow. It is a dependency among flows and parts of flows. A flow is said to be triggered if it is created or activated by another flow (e.g., the exit-flow from a queue triggers an in-flow of passengers waiting to enter). Triggering can also be used to initiate events such as starting a machine (e.g., a manager's signal triggers the opening of an additional queue to alleviate congestion).

III. Examples

IBM Maximo [21] is a widely used system in many real life environments. FM modeling is a theoretical modeling technique currently in the experimental phase. This section converts two examples of Maximo-based cases into FM descriptions. The purpose is to compare the conceptual foundations and advantages of the two process representations.

A. *Example 1*

According to IBM Maximo [21], the program is capable of producing and managing the purchasing process. Purchase requisitions and purchase orders can be generated for store items and for direct issue (to operating locations). Here, we focus on modeling the process of *purchase orders* as a sample application. Purchase orders can be generated from a work order for parts, materials, or services, request for quotation, contracts, etc. See Fig. 2 as an example of diagrammatic representation in Maximo.

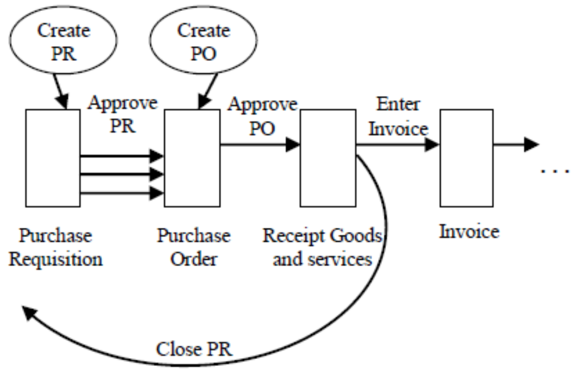

Fig. 2. The purchase process (redrawn, partial from [21]).

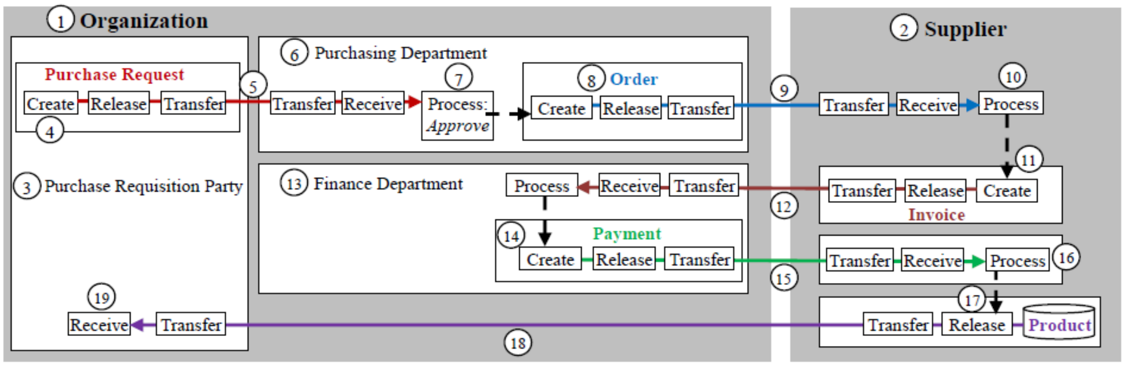

Fig. 3. FM representation of purchase process.

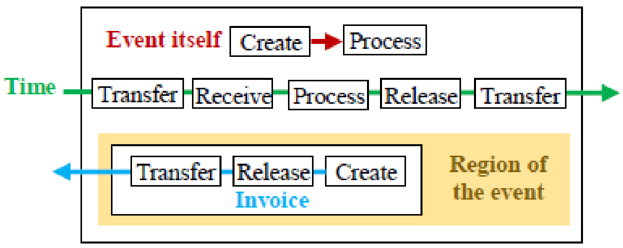

Fig. 4. The event *An invoice has been sent.*

In order to centralize purchasing all inventory purchasing might be done through a central storeroom with all other storerooms "purchasing" their stock from that central storeroom. Maximo supports this type of purchasing by allowing you to create two different types of purchase requisitions and purchase orders [21].

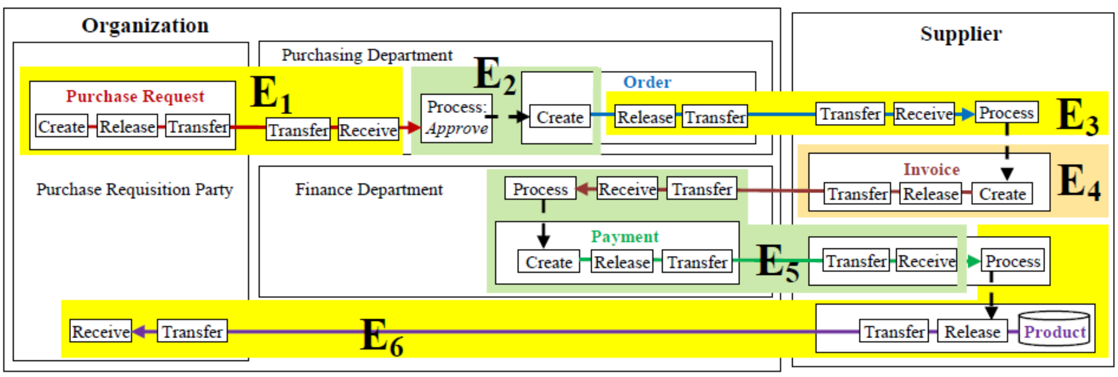

Fig. 5. Events in the FM representation of purchase process.

Fig. 3 shows the FM representation of this purchasing process, produced in accordance with our best understanding of the Maximo representation of Fig. 2. In the FM diagram of Fig. 3, there are two main spheres: the organization (circle 1) and the supplier (2). In the organization, someone, denoted as the purchase requisition party (3), creates a purchase request (4) that flows (5) to the purchasing department (6) where it is processed (7).

If approved, a purchase order is created (8) that flows to the supplier (9). The supplier processes the order (10) to trigger the creation of an invoice (11) that is sent (12) to the finance

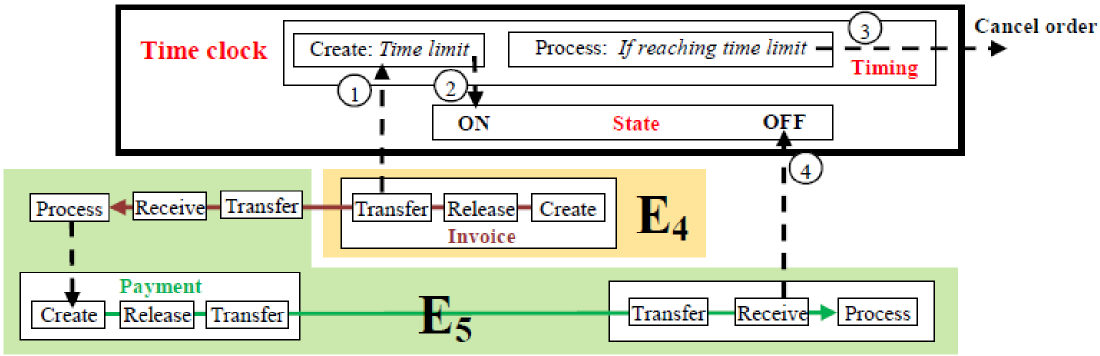

Fig. 6. A modeling sample of time constraint.

Accordingly, to build a description of the behavior of the system, six events are identified, shown in Fig. 5.

Event 1 (E_1): Purchase request is created and received.

Event 2 (E_2): The request is approved and an order is created.

Event 3 (E_3): The order is sent to and received by the supplier.

Event 4 (E_4): An invoice is sent by the supplier.

Event 5 (E_5): The invoice is received and a payment is sent by finance department.

Event 6 (E_6): After the payment is processed, the product is sent by the supplier and received by the requester.

In this example, the execution control is the sequence of events E_1 E_2 E_3 E_4 E_5 E_6. However, execution control can be much more complicated and can be used to discover concurrency, parallelism, and timing control.

Suppose that in the example, upon receiving the invoice, the customer is given a certain time to provide payment, otherwise the order is cancelled. Fig. 6 shows this case where, when the invoice is transferred, a time limit is initialized (circle 1) that turns the clock ON (2). If the time reaches its set limit, a cancellation of the order is triggered (3). Otherwise, if the payment is received then this turns the clock OFF (4).

B. *Example 2*

Here we have selected Maximo, an asset management software program, as an example to contrast with workflow modeling in FM.

In Maximo, different types of nodes can represent various points in a business process: e.g., start node, condition node, interaction node, sub-process node, task node, and stop node. A workflow process is created by inserting nodes and connection lines on a workflow canvas. There are many notions: person records, role records, action records, communication templates, notifications, escalations and action groups, etc. Also, there are many actions [22]:

- Create change, incident, problem, service request ticket, work order, etc.
- Initiate, escalate, accept, reject, etc.

A sample workflow process for handling a purchase requisition is shown in Fig. 7. Note that no attempt is made in this system to build such a flowchart-like diagram on a conceptual foundation. Only graphic symbols are introduced in flowchart fashion for modeling a process such as purchase requisition. Worse, some clients (such as the organization in our study case) believe that such a diagram can be modified for specific money values and if a step is not needed (e.g., vice president's approval) it can be passed by.

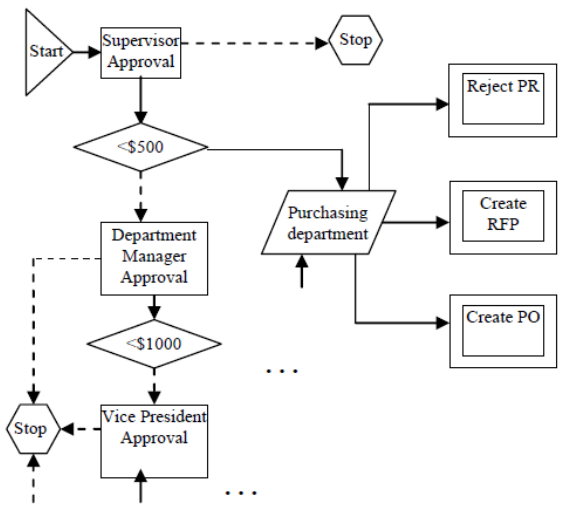

Fig. 7. A purchase requisition (redrawn, partial from [22]).

C. *FM Static Description for Handling a Purchase Requisition*

Fig. 8 describes the corresponding FM purchasing requisition for a material. In the figure, the sphere Maximo (circle 1) has seven subspheres: Employee, System, Supervisor, Department Manager, Vice President, Chief Financial Officer, and Commercial Department (circles 2, 3, 4, 5, 6, 7, and 8, respectively).

An employee creates the purchase request in (9) which in turn is transferred through the system (10) and received by the supervisor (11). Once the supervisor receives the request,

- If the employee is not authorized to place the request, a cancellation is created (12) that flows to the system (13) and creates a cancellation response (14) that is sent to the employee as a feedback (15).
- If the purchase price exceeds the limit, a rejection is created (16) that flows through the system (17) to be received by the manager (18).

- Otherwise, the request is approved (19) and directed to the commercial department (20).

The department manager and the vice president have the exact same procedure as the supervisor but the price limit differs in each of those levels. Once the manager receives the request (18),

- If the supervisor is not authorized for the request, a cancellation is created (21) that flows to the system (22) and creates a cancellation response (23) that is sent to the employee as a feedback (24).
- If the purchase price exceeds the limit, a rejection is created (25) that flows through the system (26) to be received by the vice president (27).
- Otherwise, the request is approved (28) and directed to the commercial department (29).

As for the vice president, once he receives the request (27),

- If the manager is not authorized for the request, a cancellation is created (30) that flows to the system (31) and creates a cancellation response (32) that is sent to the employee as a feedback (33).
- If the purchase price exceeds the limit, a rejection is created (34) that flows through the system (35) to be received by the chief financial officer (36).
- Otherwise, the request is approved (37) and directed to the commercial department (38).

Upon receiving the request by the chief financial officer (36), it is either cancelled (39) and a response (40) is sent to the employee (41), or approved (42) and transferred to the commercial department (43).

When an approved request reaches the commercial department (either from 20, 29, 38 or 43), a purchasing agent evaluates the record and takes one of the following actions:

- If quotation of the request is needed, a Request for Quotation (RFQ) is created (44) in order to solicit bids for the purchase request (45).
- If any of the conditions related to commercial department rules is not met, a rejection is created (46).
- Otherwise, the request is approved and a purchase order (PO) is created (47).

Any of these three actions will be created as a response by Maximo system (48) and then transferred to the employee as a feedback (49).

D. FM Dynamic Description for Handling a Purchase Requisition

To model the behavior of the purchase requisition process described in Fig. 8, we identify a set of meaningful events as shown in Fig. 9.

Event 1 (E_1): A purchase request is created by an employee and received by the system.

Event 2 (E_2): The request is processed by the system.

Event 3 (E_3): A supervisor receives the request.

Event 4 (E_4): The supervisor cancels the request.

Event 5 (E_5): The supervisor rejects the request.

Event 6 (E_6): The supervisor approves the request.

Event 7 (E_7): The cancelled request is registered in the system.

Event 8 (E_8): The system informs the employee of the cancellation.

Event 9 (E_9): The rejected request is sent to the system.

Event 10 (E_{10}): The system sends the rejected request to the department manager.

Event 11 (E_{11}): The approved request is registered in the system.

Event 12 (E_{12}): The system sends this approval to the commercial department (for simplicity, all flows to the commercial department are shown in the lower left, highlighted in yellow).

Event 13 (E_{13}): The department manager cancels the request.

Event 14 (E_{14}): The department manager rejects the request.

Event 15 (E_{15}): The department manager approves the request.

Event 16 (E_{16}): The cancelled request is registered in the system.

Event 17 (E_{17}): The system informs the employee of the cancellation.

Event 18 (E_{18}): The department manager sends the rejected request to the system.

Event 19 (E_{19}): The system sends the rejected request to the vice president.

Event 20 (E_{20}): The approved request is registered in the system.

Event 21 (E_{21}): The system sends this approval to the commercial department (yellow in lower left of diagram).

Event 22 (E_{22}): The vice president cancels the request.

Event 23 (E_{23}): The vice president rejects the request.

Event 24 (E_{24}): The vice president approves the request.

Event 25 (E_{25}): The cancelled request is registered in the system.

Event 26 (E_{26}): The system informs the employee of the cancellation.

Event 27 (E_{27}): The rejected request is sent to the system.

Event 28 (E_{28}): The system sends the rejected request to the chief financial officer.

Event 29 (E_{29}): The approved request is registered in the system.

Event 30 (E_{30}): The system sends this approval to the commercial department (yellow in lower left of diagram).

Event 31 (E_{31}): The chief financial officer cancels the request.

Event 32 (E_{32}): The chief financial officer approves the request.

Event 33 (E_{33}): The cancelled request is registered in the system.

Event 34 (E_{34}): The system informs the employee of the cancellation.

Event 35 ($E2_{35}$): The approved request is sent to the system.

Event 36 (E_{36}): The system sends this approved request to the commercial department (yellow in lower left of diagram).

Event 37 (E_{37}): The commercial department agent creates request for quotation (RFQ).

Event 38 (E_{38}): The commercial department agent rejects the request.

Event 39 (E_{39}): The commercial department agent creates purchase order (PO).

Event 40 (E_{40}): The commercial department initiates the bidding process for the RFQ.

Event 41 (E_{41}): The commercial department registers the RFQ in the system.

Event 42 (E_{42}): The commercial department registers the rejected request in the system.

Event 43 (E_{43}): The commercial department registers the PO in the system.

Event 44 (E_{44}): The system informs the employee of the decision (either E_{37}, E_{38}, or E_{39}).

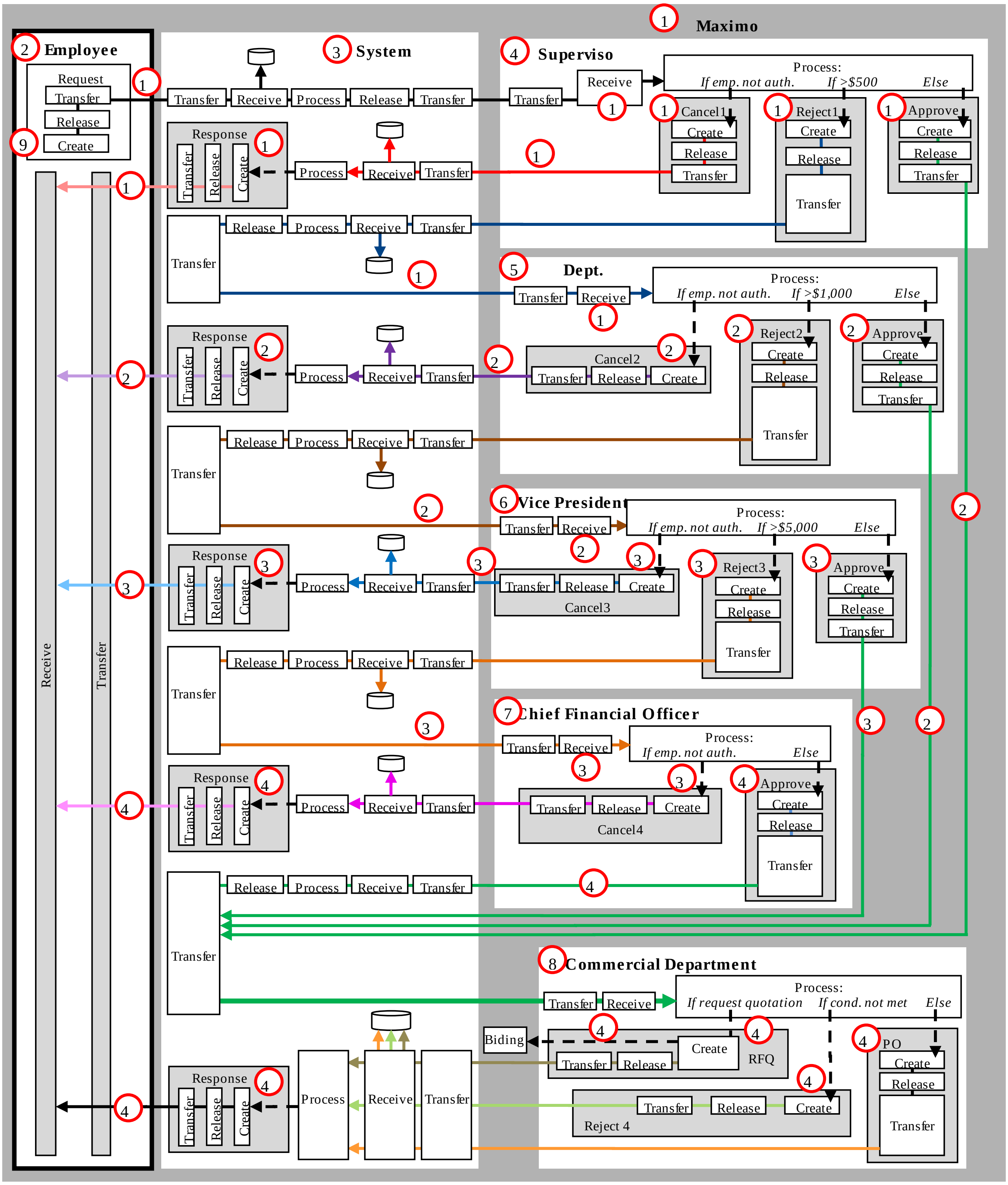

Fig. 8. FM static representation of the purchase requisition process.

Fig. 10 shows the chronology of these events. It can be utilized as a control system for many functions such as time monitoring. Fig. 11 shows a representation of the time constraint. For example, the supervisor should act on any received purchase requisition within x time units. The chronology of events can be used to respond to queries such as *What is the status of purchase requisition number 12345?* (shown in Fig. 12).

The response is that requisition number 12345 was rejected by the supervisor but approved by the department manager, who then sent it to the commercial department, where the requisition awaits further action.

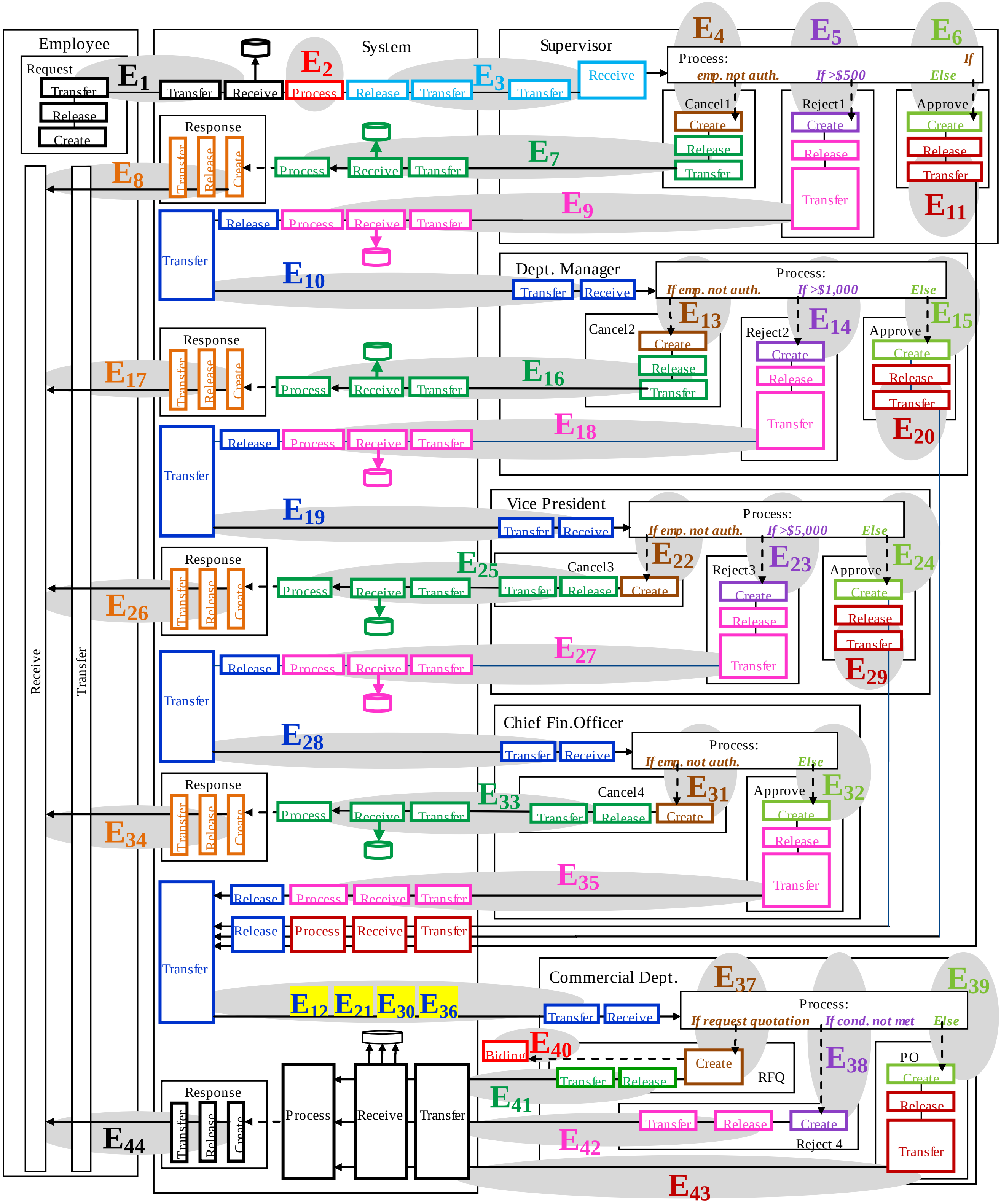

Fig. 9. FM dynamic representation of events.

IV. Case Study: Maximo Application

This model of the case study suggests the viability of FM as a diagramming tool for modeling of processes in Maximo and, in general, processes in Enterprise Asset Management. We applied FM to the process of Desktop Requisition (DTR) modeled in Maximo, basically the same concept as purchase requisition:

- An employee can create a DTR.
- A supervisor can cancel the request or approve it.
- A team leader can cancel the request or approve it.
- A manager can cancel the request or approve it.

This process is modeled in Fig. 13. In Maximo (circle 1), the process comprises six main spheres: Employee, System, Supervisor, Team Leader, Manager, and the Commercial Department (2–7). An employee creates a DTR (8) that flows to the system (9) where it is processed (10) and sent to a supervisor (11). Accordingly,

- If the employee is not authorized to create the request, a cancellation is created (12) that flows to the system (13), where a cancellation response is created (14) and sent to the employee (15).
- If any of the company's specifications are incorrect, a rejection is created (16) that flows to the system (17), where a rejection response is created (18) and sent to the employee (19). The employee has the opportunity to send a modified version (20–23), which is either rejected (24) or approved (25) by the supervisor.
- Else, the original request is approved (26).

A report initially approved by the supervisor flows to the system (27) to be sent to the requester (28) and to the team leader (29). A similar flow occurs at the team leader level until the approved request arrives at the manager (30). Finally, when the manager approves it, it flows to the commercial department (31) to enter the purchasing cycle (32).

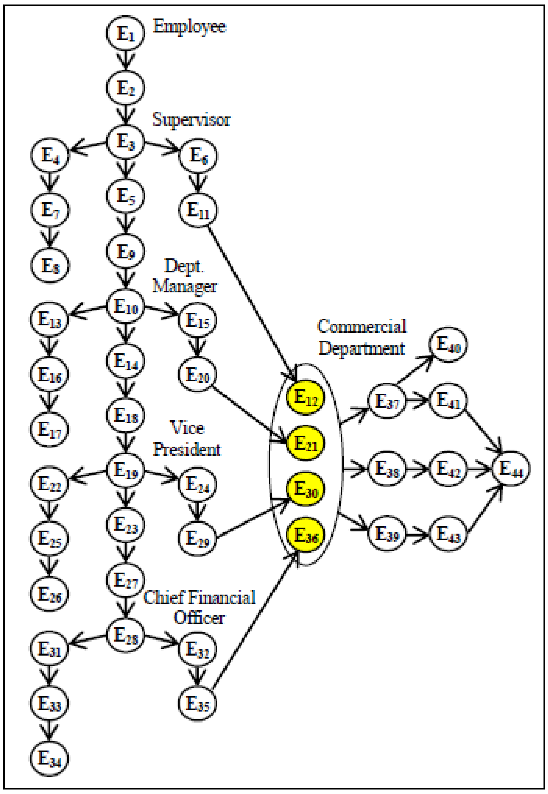

Fig. 10. Chronology of events.

The FM model can be a foundation in development of an automated system. The remarkable aspect of such a system is that it can be built on the same elements and operations as the theoretical framework. It integrates data on different platforms. System builders draw the architecture according to system modules, and screens are designed by agreement between builders and users using the same model. An FM-based system is applied uniformly through different system development phases. Its overall architecture is an FM-integrated description built upon flow machines.

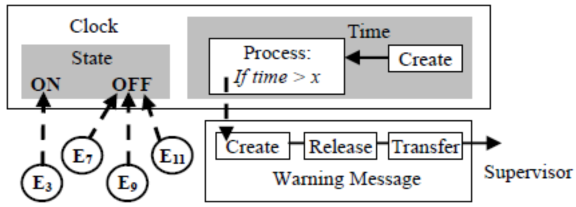

Fig. 11. Example of time control for the supervisor.

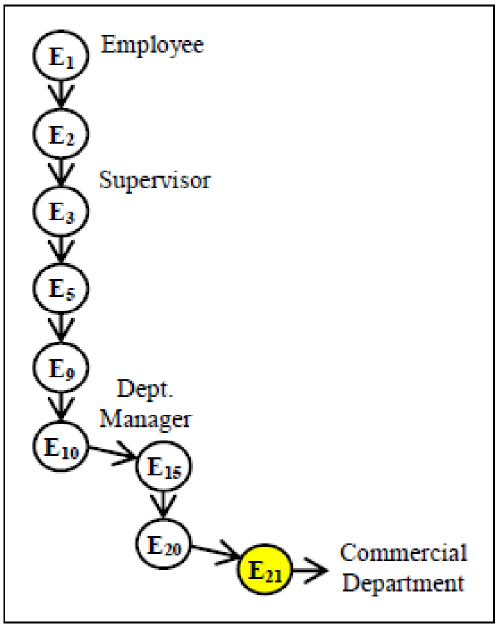

Fig. 12. Events in response to What is the status of requisition number 12345?

FM can be applied in the user interface (UI) thus introducing a unified interface instead of an ad hoc collection of icons. This holistic approach is used to describe the structure and dynamic behavior exhibited by a system, "the wholeness of the structure is preserved though extended [in the FM model], each time changing without losing the global structure" [22]. The FM diagram is a machine with an "underlying process whose steps are wholeness-extending-transformations, where each transformation operates on one wholeness [i.e., one machine] to produce another wholeness [machine] [23].

Fig. 14 shows a possible screen within the employee interface of Fig. 13. The same FM stages that are used in the design of the system are used in this interface. When the employee clicks the Create icon, a new window is displayed for entering the required information for a new desktop requisition. We have added a Process icon to facilitate changing an existing request. A Release icon would save the requisition as a finished transaction but without transferring it to the next level. Transfer causes the requisition to be sent.

Fig. 15 shows a sample of the supervisor's screen. Here we assume the supervisor has already received requisition 12345 and it is stored in the supervisor's list of requisitions, accessible by clicking the Display icon. The requisition screen allows the supervisor to handle decisions related to requisition 12345 in terms of the same FM stages.

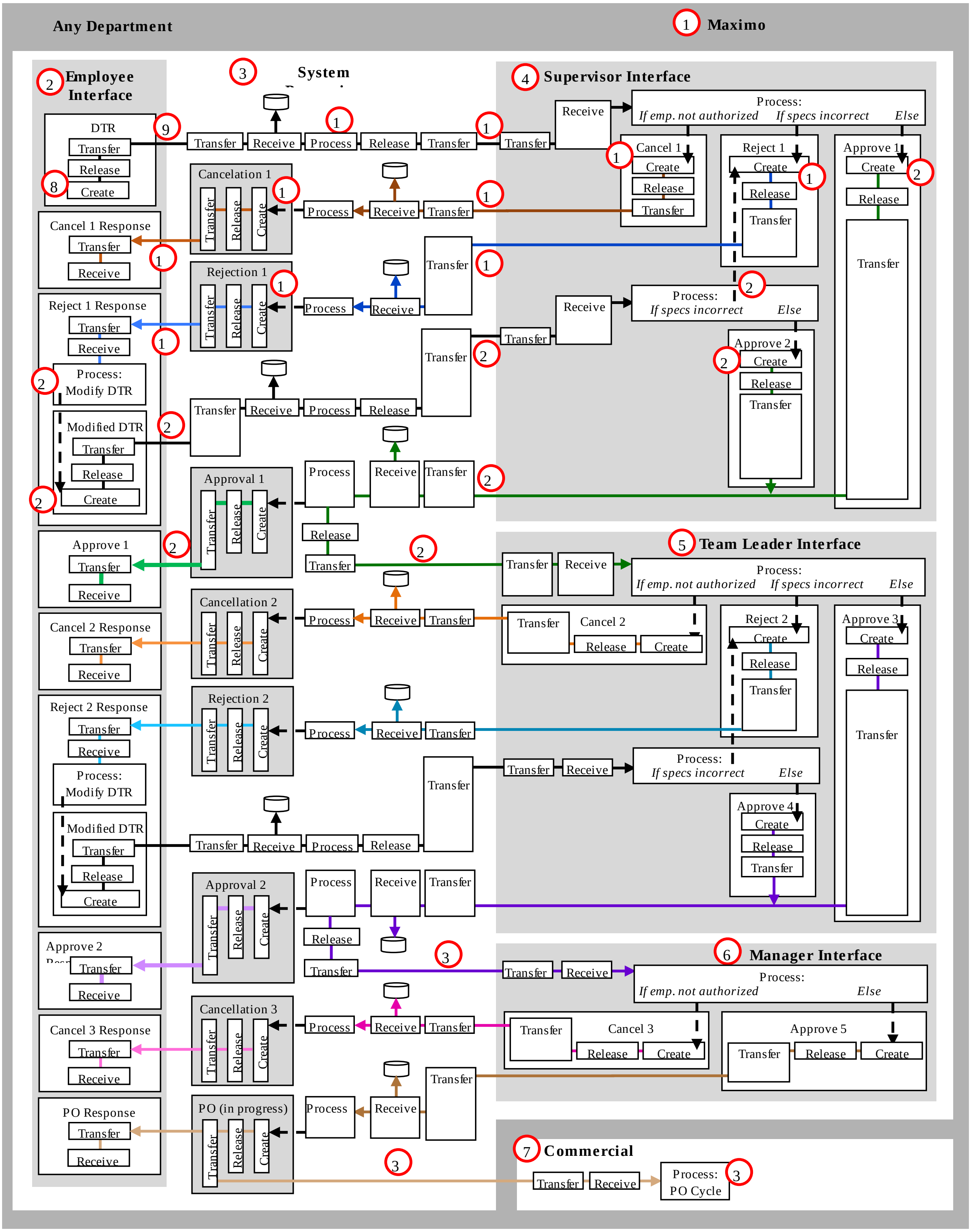

Fig. 13. FM representation of the desktop requisition case study.

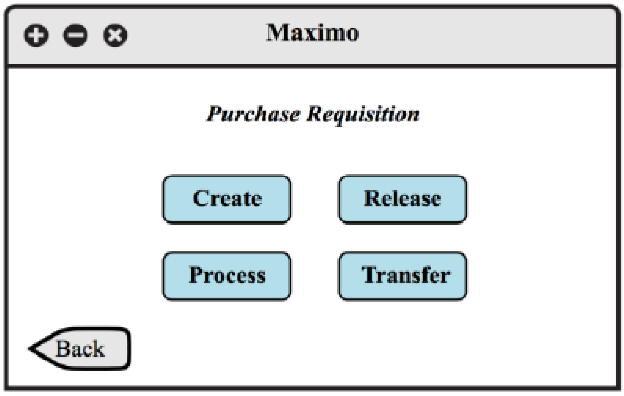

Fig. 14. Sample of the employee screen.

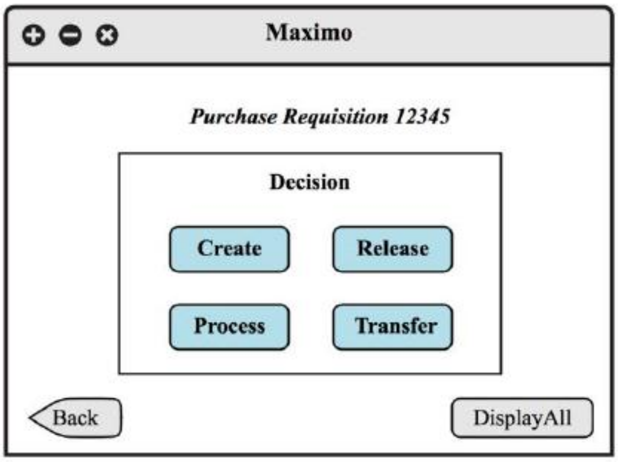

Fig. 15. Sample of the supervisor screen.

IX. CONCLUSION

Enterprise Asset Management (EAM) is a broad term for software that provides a way to view company-owned assets holistically, where the goal is to control and proactively optimize operations for quality and efficiency. This paper applies a new modeling technique, FM, built on systems of things that flow, to model EAM processes systematically. FM can be utilized uniformly at different stages of system development, from requirement specification to user interphase.

The complexity of FM diagrams may present difficulties; however, solutions to visual complexity have already been implemented in engineering systems (e.g., aircraft and high-rise building schemata) through multilevel simplifications, with details lumped together by omitting stages and unifying flows among them. Nevertheless, the underlying FM schema remains the reference for any further usages such as analysis and documentation.

Further research will work on other types of RFPs. Many issues remain to be clarified; however, this paper demonstrates potential feasibility of the approach.

The development of an EAM model using the FM modeling technique is a promising field that needs in-depth investigation. Applications of FM can be further enhanced with such features as optimization and productivity, which points to opportunities for research in use of FM in relation to EAM for more effective modeling.

REFERENCES

[1] IBM Global Business Services. *The Evolution of Asset Management*, IBM White Paper. (2007). [Online]. Available: https://www-935.ibm.com/services/uk/igs/pdf/best_practices_in_asset_management_final.pdf

[2] *Peregrine Systems, Asset Management Product Brief: The Benefits of Asset Management*, Peregrine Systems Inc., Irvine, CA.

[3] Institute of Asset Management (IAM). What is Asset Management? [Online]. Available: http://www.theiam.org.

[4] S. Lingamaneni, "Modelling and execution of asset management process via workflow automation," Thesis. School of Engineering Systems, Faculty of Built Environment and Engineering, Queensland University of Technology, Brisbane, Australia, May 2010.

[5] S. K. Kim and J. Sumberg, "Assets, asset-ness and graduation," *Int. Conf. Graduation and Social Protection*, Kigali, Rwanda, May 2015.

[6] A. Dorward, S. Anderson *et al.*, (2001). "Asset functions and livelihood strategies: A framework for pro-poor analysis," *Policy Pract,* vol. 1, 2001.

[7] TechTarget, Enterprise Asset Management (EAM) [Online]. Available: http://searcherp.techtarget.com/definition/enterprise-asset-management-EAM

[8] N. Shetty, A. Hayes, D. Pocock, and J. Watts. Leveraging the relationship between BIM and asset management, document produced jointly by institution of civil engineers (ICE), chartered institution of civil engineering surveyors (ICES), and institute of asset management (IAM). [Online]. Available: https://theiam.org/sites/default/files/upload/bimmodellingandassetmanagementpositionpaperv2bg.pdf

[9] Institute of asset management. (IAM), Asset information model AIM. [Online]. Available: https://www.designingbuildings.co.uk/wiki/Asset_information_model_AIM

[10] A. Koronios, D. Nastasie, V. Chanana, and A. Haider, "Integration through standards–an overview of international standards for engineering asset management," in *Proc. Fourth Int. Conf. Condition Monitoring*, Harrogate, UK, June 2007.

[11] J. Black. How to prepare better RFP requirements lists for IT success. [Online]. Available: http://blog.fhblackinc.com/how-to-prepare-better-rfp-requirements-lists

[12] Y. Qi, "The modeling and implementation of the asset management system of the automobile maintenance shop based on RFID," in *Proc.2016 Int. Conf. Education, Management Science and Economics (ICEMSE-16)*, pp. 407-409.

[13] S. Al-Fedaghi and Y. Alduwaisan, "Modeling of an enterprise and information system: process specification based on the flow of things," in *Proc. Int. Conf. Software and Services Engineering (ICSSE 2018)*, April 2018.

[14] S. Al-Fedaghi and O. Alsumait, **"**Writing security specification with things that flow," presented at *2017 Int. Conf. Security with Intelligent Computing and Big-data Services*, Hualien, Taiwan, Dec. 15-17, 2017.

[15] S. Al-Fedaghi and H. Alahmad, **"**Integrated modeling methodologies and languages," presented at *ACM 12th Int. Conf. Ubiquitous Information Management and Communication*, Langkawi, Malaysia, Jan. 5-7, 2018.

[16] S. Al-Fedaghi and H. Alahmad, "Orientation in conceptual modeling frameworks," presented at *3rd IEEE Int. Conf. Big Data Intelligence and Computing*, Orlando, USA, Nov. 6-10, 2017.

[17] S. Al-Fedaghi and A Abdullah, "Flow-based systems modeling: application in airport system description,: *Int. J. Ind. Syst. Eng.*, vol. 25, no. 3, pp. 318–334, 2017.

[18] S. Al-Fedaghi, "Software engineering modeling applied to English verb classification (and poetry)," *Int. J. Comput. Sci. Inform. Secur.*, vol. 15, no. 10, 2017.

[19] S. Al-Fedaghi, "Toward understanding dynamic behavior: events and states in abstract machines," *J. Knowl. Syst. Sci. (IJKSS)*, vol. 8, no. 4, 2017.
[20] IBM Maximo, Maximo core concepts. [Online]. Available: https://www.slideshare.net/sunk818/ibm-maximo-keyconceptsv4
[21] C. Alexander, *The Nature of Order: The Process of Creating Life*. Taylor & Francis. (2002).
[22] A. Obtulowicz, "Some mathematical methods and tools for an analysis of harmony-seeking computations," in *Proc. Fifth Brainstorming Week on Membrane Computing*, M. A. Gutierrez-Naranjo, Gh. Paun, A. Romero-Jimenez, and A. Riscos-Nunez (eds.), Sevilla, Spain, Jan. 29–Feb. 2, 2007, pp. 246-262.

Sabah Al-Fedaghi is an associate professor in the Department of Computer Engineering at Kuwait University. He holds an MS and a PhD from the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, Illinois, and a BS in Engineering Sciences (computer) from Arizona State University. He has published more than 290 journal articles and papers in conferences on Conceptual modeling, Software Engineering, Database Systems, information Ethics, Privacy and Security. He previously worked as a programmer at the Kuwait Oil Company and headed the Electrical and Computer Engineering Department (1991–1994) and the Computer Engineering Department (2000–2007).

Nourah Al-Huwais is a computer engineer in the Information Technology Department at Kuwait National Petroleum Company. She holds a BS in Computer Engineering from the Computer Engineering Department, Kuwait University. And is a graduate student from the same University at the current time. She is certified in CAPM, CCNA, and COBIT5.